

EL LENGUAJE SQL

Elmasri y Navathe 2007

*Cap 8 "SQL 99: definición del esquema,
restricciones, consultas y vistas"*

Índice

◆ Introducción

- ◆ LMD: consultas
- ◆ LDV
- ◆ LMD: actualizaciones
- ◆ LDD
- ◆ LMD: más consultas

SQL: Structured Query Language

- ◆ Lenguaje estándar
- ◆ Los SGBD comerciales utilizan variantes
- ◆ Revisiones:
 - SQL1 1986
 - SQL2 1992
 - SQL-1999
 - SQL-2003
 - SQL-2006
 - SQL-2008
 - SQL-2011
 - SQL-2016



SQL lenguaje declarativo

- ◆ Se describe el resultado esperado
- ◆ El SGBD elige:
 - Qué operaciones usa
 - En qué orden
- ◆ Esto último es la optimización

Lenguajes en SQL

- ◆ **LDD**: de definición de datos
- ◆ **LMD**: de manipulación de datos
 - Incluye consulta y actualización
- ◆ **LDV**: de definición de vistas

Otros lenguajes en SQL

- ◆ Especificación de seguridad y autorización
- ◆ Especificación de control de transacciones
- ◆ Reglas para inclusión en lenguajes de programación (C, Java...)

Índice

- ◆ Introducción
- ◆ **LMD: consultas**
- ◆ LDV
- ◆ LMD: actualizaciones
- ◆ LDD
- ◆ LMD: más consultas

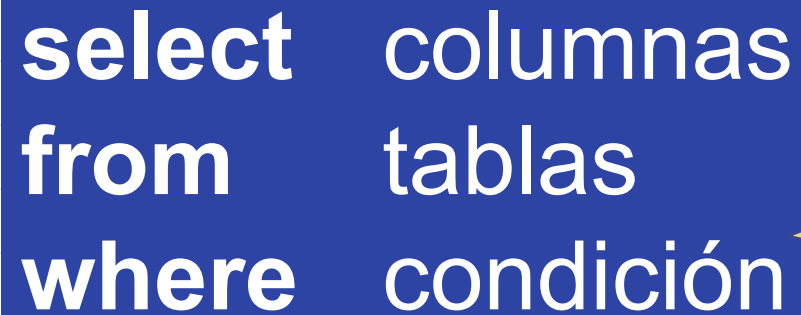
Índice / consultas

1. Sobre una tabla

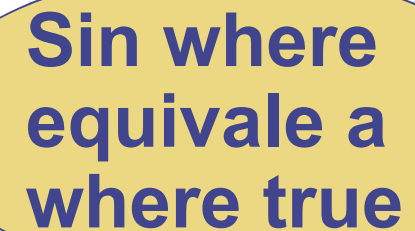
2. Join

3. Group by y having

Consultas básicas



select columnas
from tablas
where condición



Sin where
equivale a
where true

El resultado puede contener
TUPLAS REPETIDAS

Select all

cliente

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	Haro
Jorge	Sáenz	Logroño
Rosa	Díaz	Logroño
Juan	Sánchez	Madrid

```
select all nom, apell  
from cliente
```

nom	apell
Juan	Sánchez
Luis	Ramírez
Rosa	Díaz
Jorge	Sáenz
Rosa	Díaz
Juan	Sánchez

Sólo select = select all

cliente

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	Haro
Jorge	Sáenz	Logroño
Rosa	Díaz	Logroño
Juan	Sánchez	Madrid

```
select nom, apell  
from cliente
```

nom	apell
Juan	Sánchez
Luis	Ramírez
Rosa	Díaz
Jorge	Sáenz
Rosa	Díaz
Juan	Sánchez

Select distinct

cliente

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	Haro
Jorge	Sáenz	Logroño
Rosa	Díaz	Logroño
Juan	Sánchez	Madrid

```
select distinct nom, apell  
from cliente
```

nom	apell
Juan	Sánchez
Luis	Ramírez
Rosa	Díaz
Jorge	Sáenz
Rosa	Díaz
Juan	Sánchez

¡ Fíjate, sólo era un distinct !

```
select distinct (nom), distinct (apell)  
from cliente
```

**¡Error
común!**

¿Por qué aquí sí funciona?

```
select distinct (nom), apell  
from cliente
```

```
select distinct (nom), (apell)  
from cliente
```

```
select distinct ((nom), (apell))  
from cliente
```

Select *

cliente

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	Haro
Jorge	Sáenz	Logroño
Rosa	Díaz	Logroño
Juan	Sánchez	Madrid

select * from cliente

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	Haro
Jorge	Sáenz	Logroño
Rosa	Díaz	Logroño
Juan	Sánchez	Madrid

Select con +, -, *, / ó %

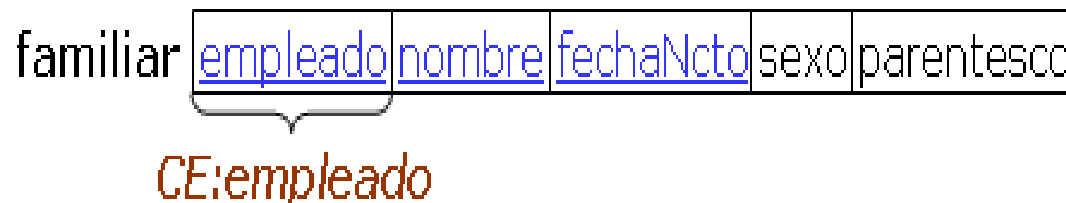
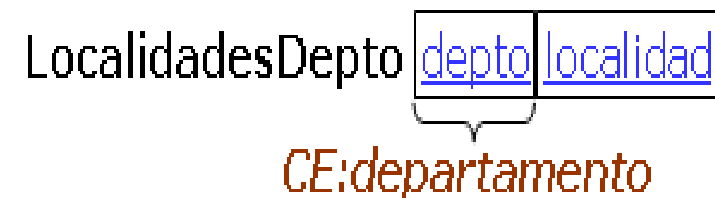
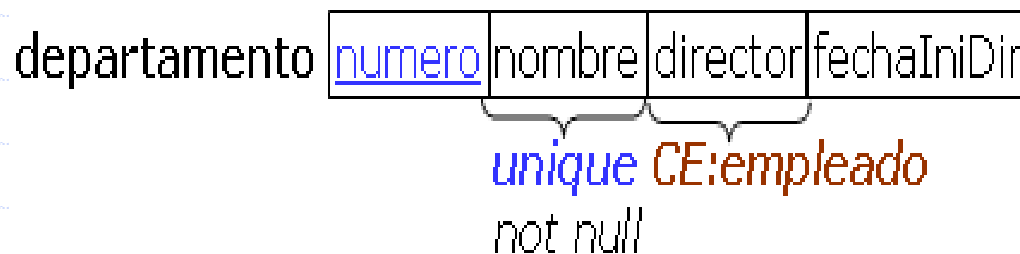
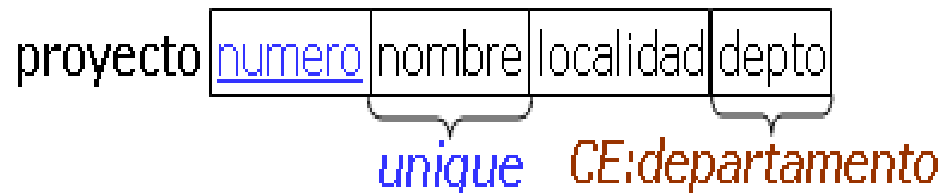
empleado

nom	apell	salario
Pepe	Martín	1000
Lola	Balda	2500
Iker	Pereda	800
Gema	Prat	5000

```
select apell, salario*1.1  
from empleado
```

apell	salario*1.1
Martín	1100
Balda	2750
Pereda	880
Prat	5500

Empresa



unique = clave candidata

Bibliotecas

libro id titulo editorial año peso paginas precio primer_solicitante
CE:editorial *CE:lector*

editorial nombre direccion tfno

autor id nombre apellidos

autorlibro autor libro
CE:autor *CE:libro*

lector id nombre apellidos direccion tfno fecha_inscripcion

biblioteca id nombre direccion tfno

ejemplar libro ejemplar biblioteca fecha_solicitud fecha_compra
CE:libro *CE:biblioteca*
not null

prestamo libro ejemplar fecha_prestado fecha_devuelto lector
CE:ejemplar *CE:lector*
not null

Comercio

cliente	<u>ident_client</u>	apell1	nombr_client	nacion	ciud
	not null				

tienda	<u>ident_tiend</u>	ciud	nombr_gerent
	not null		

proveedor	<u>ident_prov</u>	nombr_prov
	not null	

articulo	<u>ident_artic</u>	nombr_artic	peso	color	prec_compr	prec_venta	ident_prov
	not null				not null	not null	CE:proveedor

venta	<u>ident_client</u>	<u>ident_tiend</u>	<u>ident_artic</u>	<u>fech_venta</u>	prec_total	cant_artics
	CE:cliente	CE:tienda	CE:articulo			

unique=clave candidata

Universidad

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

unique

not null

not null

CE:profesor
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura

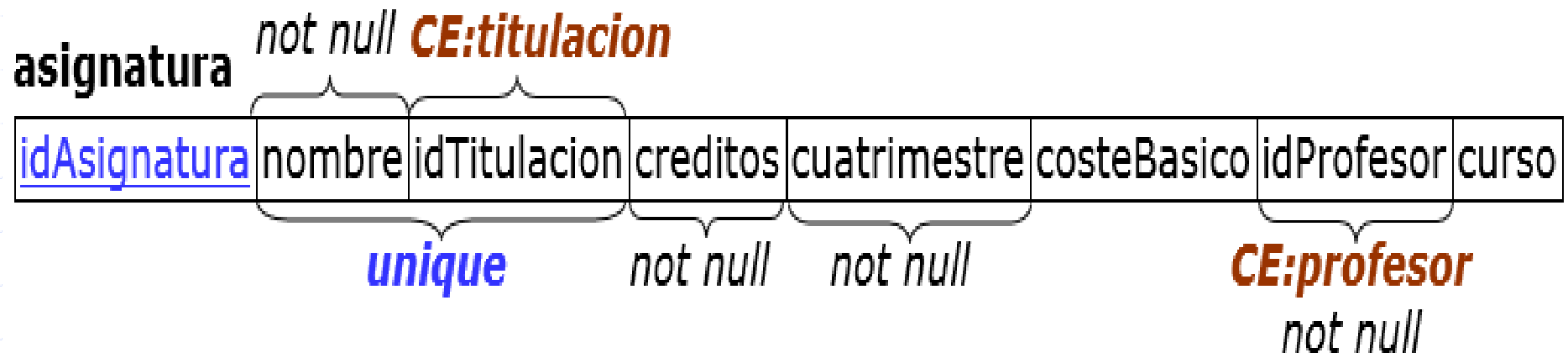
*Nº de veces que
se ha matriculado*

unique = clave candidata

Ejercicios con una tabla

- 101. Nombre y nº de horas de cada asignatura (1 crédito = 10 horas)
- 102. Nombre y coste básico por hora de cada asignatura

Ejercicios 1, 2, 3, 4



Con where

cliente

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	Haro
Jorge	Sáenz	Logroño
Rosa	Díaz	Logroño
Juan	Sánchez	Madrid

```
select *  
from cliente  
where nom='Jorge'
```

nom	apell	local
Jorge	Sáenz	Logroño

Condiciones en where

◆ Comparaciones: { =, <>, <, >, <=, >= }

◆ Lógicos: NOT, AND, OR

◆ Prioridad: NOT > AND > OR
y paréntesis

Fíjate cómo se
pone **distinto de**

Where con +, -, *, / ó %

empleado

nom	apell	salario
Pepe	Martín	1000
Lola	Balda	2000
Iker	Pereda	800
Gema	Prat	5000

```
select, apell, nom  
from empleado  
where salario / 2 >= 1000
```

apell	nom
Balda	Lola
Prat	Gema

Literales en Oracle y MySQL

◆ '...' para caracteres y fechas

◆ Formato fecha:

- DD-MON-YYYY (**Oracle**)

- YYYY-MM-DD (**MySQL**)

Cambiar:
*alter session set
nls_date_format =
'DD-MON-YYYY'*

Cambiar:
DATE_FORMAT
(date, format)

Literales en Access

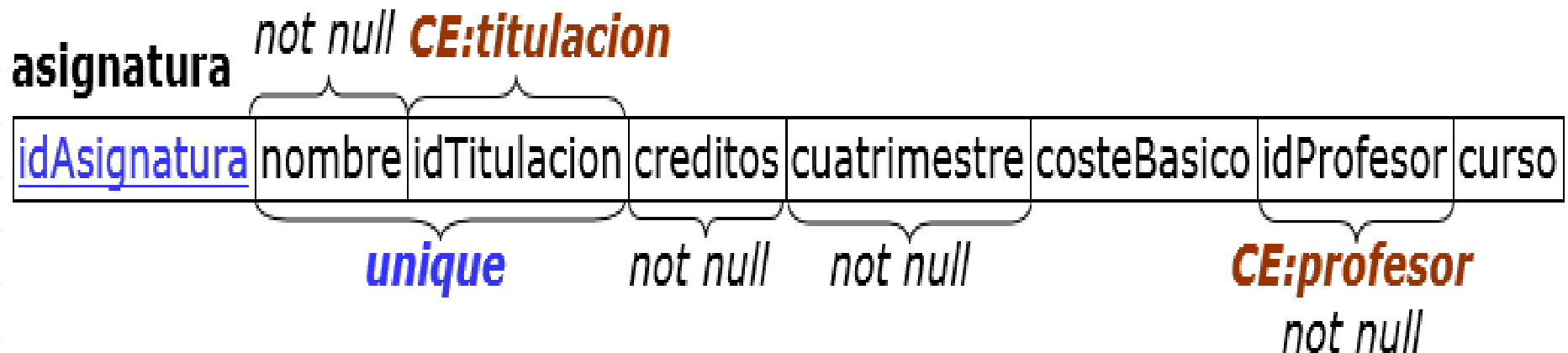
◆ "..." caracteres

◆ #...# fechas

Ejercicios con where

- 103. Nombre de las asignaturas de 1^{er} cuatrimestre con 4.5 ó 9 créditos
- 104. Nombre y código de las asignaturas del 1^{er} cuatrimestre ó con 4.5 créditos, excepto las que cumplan ambas cosas

Ejercicios 14 y 15



Algunas funciones de fecha

- ◆ `current_date()` → fecha actual
- ◆ `year('2015-02-20')` → 2015
- ◆ `month(f)`
- ◆ `day(f)`
- ◆ `datediff (f1,f2)` → f1-f2 en días
- ◆ `adddate(f, 2)` → f más 2 días
- ◆ `subdate(f, 2)` → f menos 2 días

Ejercicios con fechas

27. Nombre y edad de las personas con más de 20 años
28. Nombre y dni de las personas que cumplen años mañana

persona

<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

Funciones de texto

◆ `left('hola',2)` → 'ho'

◆ `right('hola',2)` → 'la'

◆ `length('hola')` → 4

◆ `substring('abcde',2)` → 'bcde'

◆ `substring('abcde',2,3)` → 'bcd'

◆ `instr('abcde','cd')` → 3

Más funciones

◆ `concat('abc','de',...)` → 'abcde...' (concatena)

◆ `Exp1 || exp2` → también concatena

◆ `truncate(3.45, 1)` → 3.4

En
Access
con &

Alias en select (as)

```
select apell || ', ' || nom as cliente  
from cliente
```

única
columna
de nombre
cliente

Ejercicios con funciones

105. Nombre, apellido y dni de los nacidos antes del 75 que **NO** viven en 'Madrid'
106. Nombre y apellido de aquellos de 'Logroño' cuyo teléfono no empieza por '941'
107. Nombre, apellidos y dirección de todas las personas. La dirección en una sola columna llamada 'dirección'

persona

<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

Ejercicios con funciones

108. Ciudad de los lectores inscritos los últimos 30 días sin repeticiones. La columna debe llamarse 'ciudad'. Todas las direcciones están escritas así:
'C/ Gran Vía 17, Logroño'
'Avd Portugal 3, Arnedo' ...

lector	<u>id</u>	nombre	apellidos	direccion	tfno	fecha_inscripcion
---------------	-----------	--------	-----------	-----------	------	-------------------

Is null / is not null

nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	null
Jorge	Sáenz	Logroño
Luisa	Díez	null
Juan	Sánchez	Madrid

```
select *  
from cliente  
where loc = null
```

nom	apell	loc
-----	-------	-----

Solución vacía
¿¿Por qué??

¿Qué da algo operado con null?

where loc = null

equivale
a false

select salario + null

null

select nom || null

nom

Con is null / is not null

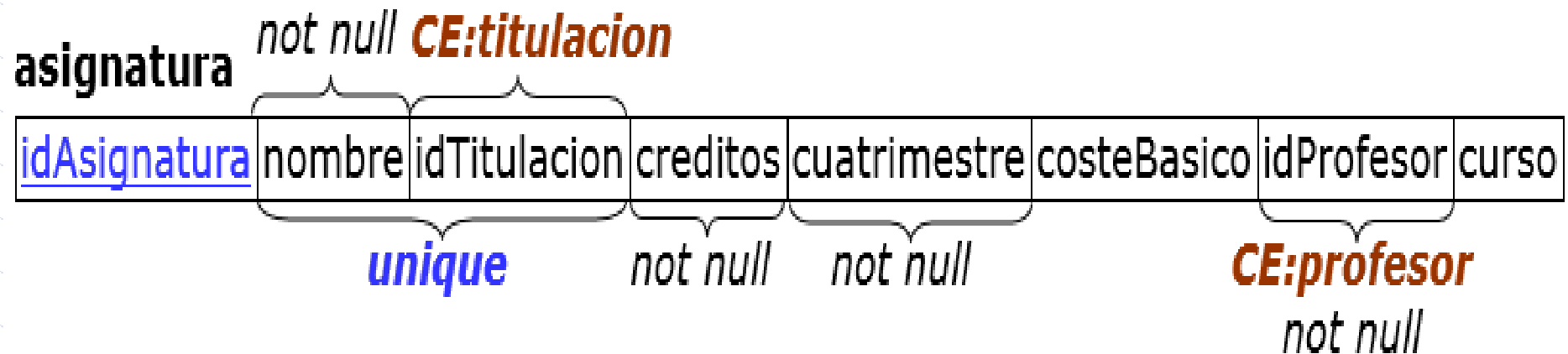
nom	apell	loc
Juan	Sánchez	Bilbao
Luis	Ramírez	Haro
Rosa	Díaz	null
Jorge	Sáenz	Logroño
Luisa	Díez	null
Juan	Sánchez	Madrid

```
select *  
from cliente  
where loc is null
```

nom	apell	loc
Rosa	Díaz	null
Luisa	Díez	null

Ejercicio con is null

25. Asignaturas que no son de ninguna titulación



Order by con asc y desc

cliente

fNac	apell	loc
2005	Sánchez	Bilbao
1998	Ramírez	Haro
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño
2002	Sánchez	Logroño

```
select fNac, apell
from cliente
where loc = 'Logroño'
order by fNac desc, apell
```

fNac	apell
2004	Díez
2002	Sáenz
2002	Sánchez

Ordena filas
des= descendente

Ejercicio ordenación/1

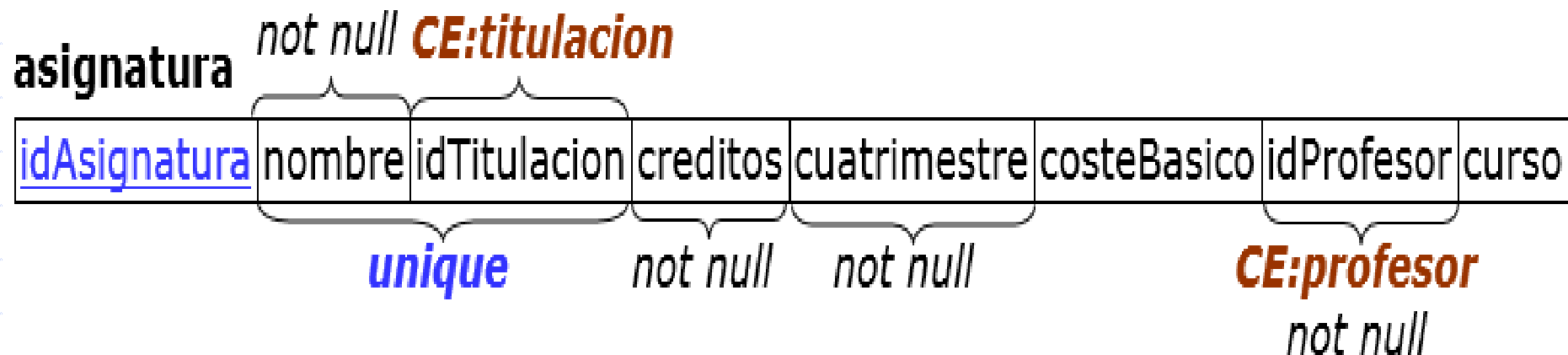
29. Listado de personas ordenadas por apellidos y nombre

persona

<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

Ejercicio ordenación/2

109. Id de titulación, coste básico, nombre y créditos de las asignaturas de 1º cuatrimestre, con profesor asignado y de entre 6 y 12 créditos. Mostrarlas ordenadas por idTitulación creciente y coste decreciente



Comparaciones between

cliente

fNac	apell	loc
2005	Sánchez	Bilbao
1998	Ramírez	Haro
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño
2002	Sánchez	Logroño

```
select fNac, apell
from cliente
where fNac between
      2000 and 2002
```

fNac	apell
2000	Díaz
2002	Sáenz
2002	Sánchez

Comparaciones in/not in

cliente

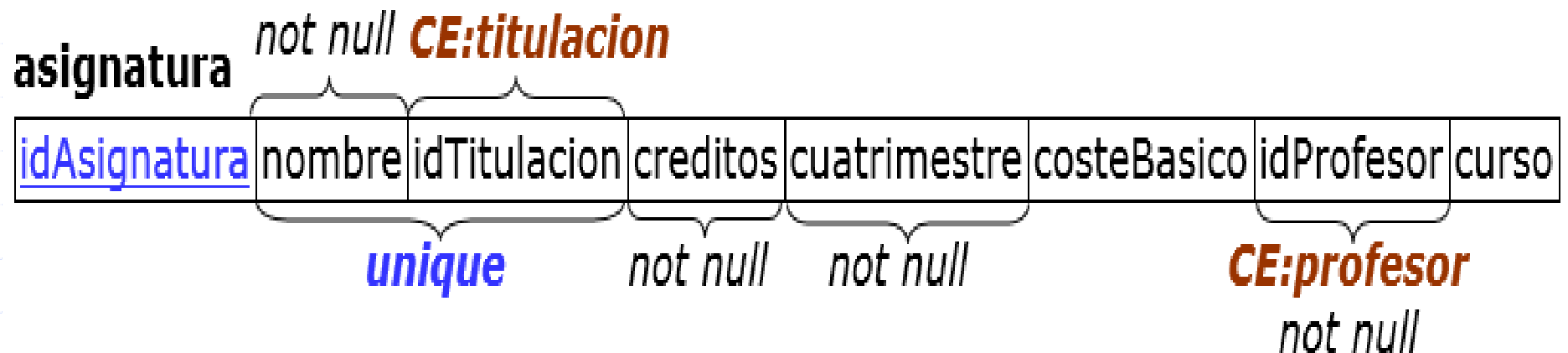
fNac	apell	loc
2005	Sánchez	Bilbao
1998	Ramírez	Haro
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño
2002	Sánchez	Logroño

```
select fNac, apell
from cliente
where fNac in (2002,
2000, 2001)
```

fNac	apell
2000	Díaz
2002	Sáenz
2002	Sánchez

Ejercicios between, in

16. Nombre de las asignaturas con coste básico entre 25 y 35 junto a las que no tienen todavía coste básico ordenadas por coste básico



Ejercicios between, in

17. Id de los alumnos (sin repeticiones) matriculados en alguna de las siguientes asignaturas: 1502, 1301, 2014 o 1590

alumnoAsignatura	<u>idAlumno</u>	<u>idAsignatura</u>	numeroMatricula
	CE:alumno CE:asignatura		Nº de veces que se ha matriculado

Comparaciones con [not] like, % y _

cliente

fNac	apell	loc
2005	Sánchez	Bilbao
1998	Ramírez	Haro
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño
2002	Sánchez	Logroño

```
select apell  
from cliente  
where apell like '%ez'
```

apell
Sánchez
Ramírez
Díez
Sánchez

Patrones para like / not like

- ◆ Tras **like** va un patrón entrecomillado
- ◆ Dos comodines:
 - **_** → equivale a 1 carácter
 - **%** → 0, 1 o más caracteres

Access
usa ? Y *

cliente

fNac	apell	loc
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño

```
select apell  
from cliente  
where apell like 'Dí_z'
```

apell
Díaz
Díez

No uses like / not like para...

comparaciones sin patrón

```
select apell  
from cliente  
where apell like 'Sánchez'
```



```
select apell  
from cliente  
where apell = 'Sánchez'
```


Ejercicios con like / not like

- 22. Nombre y apellido de los que tengan apellido que empieza por G
- 110. Apellido y fecha de nacimiento de los nacidos entre los 70 y los 90 con apellido compuesto (estilo Díez **de** Gereñu) cuyo prefijo no sea el de Álava (945) ordenados por apellido y nacimiento

persona

<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

Comparación en bloque (=)

cliente

fNac	apell	loc
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño

```
select apell, fNac
from cliente
where (apell, fNac) =
      ('Díez', 2004)
```

apell	fNac
Díez	2004

Equivale a:
apell='Díez'
and fNac=2004

Comparación en bloque (<, >...)

cliente

fNac	apell	loc
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño

```
select apell, fNac
from cliente
where (apell,fNac) <
('Díez',2004)
```

apell	fNac
Díaz	2000

Equivale a:
apell<'Díez' or
(apell='Díez'
and fNac<2004)

Coalesce

Función que devuelve el **primer NO nulo** de la lista

2ª
columna

hacienda

<u>nom</u>	salario	paro	salSocial
Matías	null	null	20.000
Marta	null	30.000	null
Maider	40.000	null	null

```
select nombre,  
coalesce(salario,  
          paro,  
          salSocial )  
as sueldo  
from hacienda;
```

en
Access
es NZ

nom	sueldo
Matías	20.000
Marta	30.000
Maider	40.000

Case

Generaliza coalesce permitiendo hacer comparaciones más allá de nulos.

En los **when** cualquier condición (AND, OR, ...)

select P.nombre,

2^a
columna

case

when P.estadoCivil='S' **then** 'Soltero/a'

when P.estadoCivil='C' **then** 'Casada/o'

when P.estadoCivil='D' **then** 'Divorciado/a'

else 'Viuda/o'

end as 'Estado civil',

P.fechaNcto

from Persona P;

en Access
con switch
e iif

Case sobre atributo/expresión

Condiciones de igualdad sobre el atributo/expresión
No compara bien con null (necesitaría is [not] null)

```
select P.nombre,  
       case P.estadoCivil  
         when 'S' then 'Solero/a'  
         when 'C' then 'Casada/o'  
         when 'D' then 'Divorciado/a'  
         else 'Viuda/o'  
       end,  
       P.fechaNcto  
from   Persona P;
```

Ejercicios con coalesce y case

111. Código y nombre de cada asignatura junto a la palabra 'barata' si su precio por crédito es menor de 5€, 'normal' si está entre 5€-10€ (ambos inclusive) y 'cara' si es mayor de 10€.

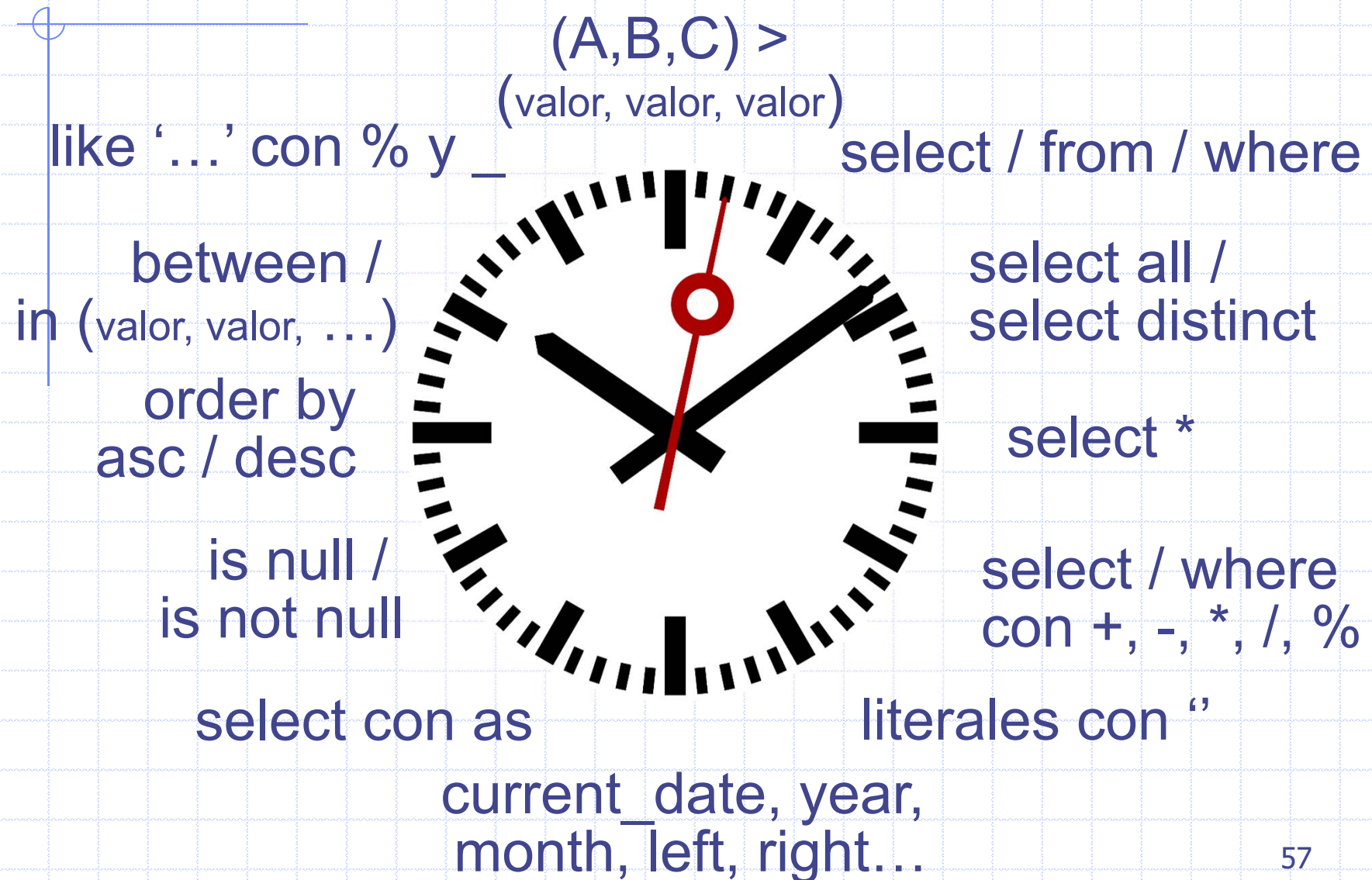


Ejercicios con coalesce y case

112. Para cada asignatura obtener su código, nombre e identificador de titulación. Si no tiene titulación poner 'Asignatura de Campus' en lugar del código de titulación.



SQL básico



Funciones agregadas

cliente

fNac	apell	loc
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño

```
select max(fNac)
from cliente
where loc = 'Logroño'
```

max(fNac)

2004

Sólo una
fila
resultado

Se pueden poner varias funciones

cliente

fNac	apell	loc
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño

```
select max(fNac),  
       min(apell),  
       count(loc)  
from cliente  
where loc = 'Logroño'
```

max..	min..	count..
2004	Díez	2

Recuerda:
así devuelve
sólo una fila

Funciones sin atributos 'suelto'

cliente

fNac	apell	loc
2000	Díaz	Haro
2002	Sáenz	Logroño
2004	Díez	Logroño

```
select max(fNac), apell  
from cliente  
where loc = 'Logroño'
```

ERROR

No se pueden
poner atributos
así en select

¿Qué funciones agregadas hay?

◆ Sobre una columna/expresión numérica:

- **sum**: calcula la suma
- **avg**: calcula la media

◆ Sobre cualquier columna/expresión:

- **min**: calcula el mínimo
- **max**: calcula el máximo
- **count**: cuenta filas con la colum/expr no nula
- **count(*)**: cuenta filas

Funciones agregadas y nulos

expresión:
salario+10

empleado

nombre	salario	dpto
Juan	null	5
Felisa	10.000	null
Ramón	10.000	5
Abel	20.000	4
Iván	20.000	1

```
select count(dpto),  
       count(*),  
       sum(salario+10),  
       avg(salario)  
from empleado
```

co..	co..	su..	av..
4	5	60.040	15.000

Los nulos se ignoran y en
count(*) no intervienen

Más sobre funciones y nulos

empleado

nombre	salario	dpto
Juan	null	5
Felisa	null	null
Ramón	null	5
Abel	null	4
Iván	null	1

```
select count(salario),  
       sum(salario),  
       avg(salario)  
from empleado
```

co..	su..	av..
0	null	null

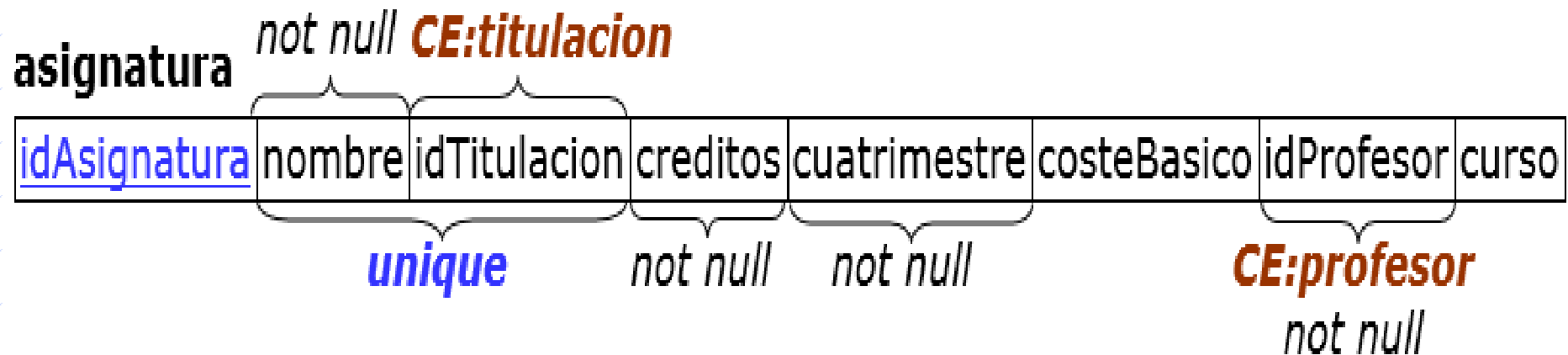
empleado

nombre	salario	dpto
--------	---------	------

co..	su..	av..
0	null	null

Ejercicios func. agregadas/1

6. Suma y media del coste básico incrementado un 10% de las asignaturas
10. Créditos y costes máximo, mínimo y medio de las asignaturas. Poner a las columnas nombres apropiados



Ejercicios func. agregadas/2

113. Cuántos días hay entre los nacimientos del más joven y el más viejo de la BD

persona

<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

Funciones con distinct dentro

Access no
lo tiene

empleado

nombre	salario	dpto
Juan	null	5
Felisa	10.000	null
Ramón	10.000	5
Abel	20.000	4
Iván	20.000	1

¡ **distinct** dentro
del paréntesis !

```
select  
  count(distinct dpto),  
  sum(distinct salario),  
from empleado
```

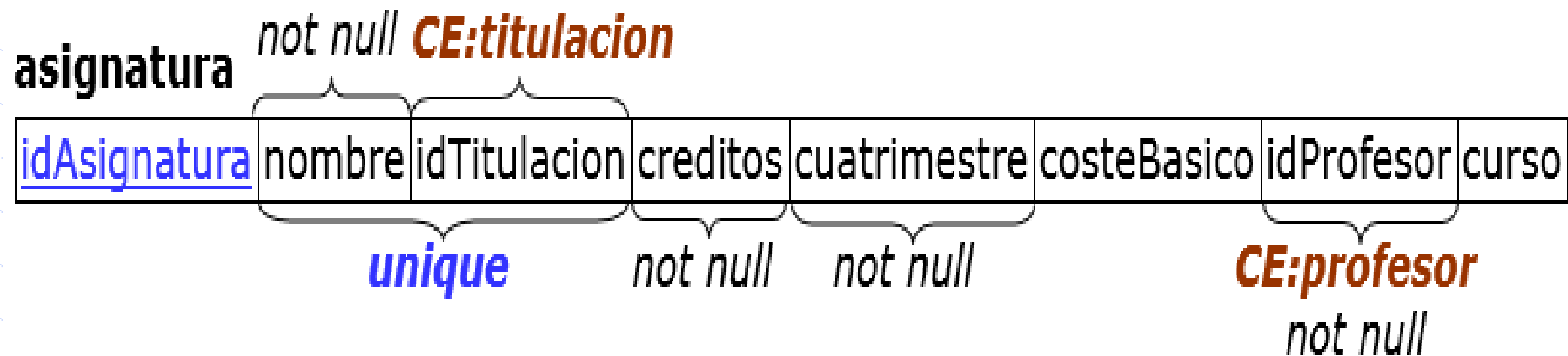
co..	su..
3	30.000

Muy diferente a
select distinct

Ejercicios func. agregadas/3

114. Cuántas asignaturas tienen coste básico nulo y cuántos costes básicos distintos hay

Ejercicios 7 y 12



Funciones agregadas y case/coalesce

Número de subordinados de los empleados con NSS 25 y 99.

```
select  
  count(case when jefe='25' then 1 end) as '25',  
  count(case when jefe='99' then 1 end) as '99'  
from empleado where jefe in ('25','99')
```

empleado

NSS	Nombre	Jefe
32	Pepe	25
25	Felisa	null
22	Ramón	99
11	Abel	25
99	Iván	25

25	99
3	1

Ejercicios func. agregadas y case/coalesce

115. Quién invierte más horas en proyectos con códigos entre 10 y 500 ¿el empleado con NIF 55 o el 99?

trabaja

empleado	proyecto	horas
----------	----------	-------

CE:empleado CE:proyecto

Funciones agregadas



max(A)

min(A)

count(A)

count(*)

sum(A)

avg(A)

estos dos
devuelven
lo mismo
que sin
distinct

max(distinct A)

min(distinct A)

count(distinct A)

sum(distinct A)

avg(distinct A)

Índice / consultas

1. Sobre una tabla

2. Join

3. Group by y having

From con varias tablas

Producto
cartesiano

dep

<u>num</u>	nomb
1	DMC
4	LSI
5	ATC

dep_loc

<u>dep</u>	<u>loc</u>
1	Logroño
5	Arnedo

CE:dep

```
select *  
from dep, dep_loc
```

num	nomb	dep	loc
1	DMC	1	Logroño
1	DMC	5	Arnedo
4	LSI	1	Logroño
4	LSI	5	Arnedo
5	ATC	1	Logroño
5	ATC	5	Arnedo

Producto cartesiano explícito

dep

<u>num</u>	nomb
1	DMC
4	LSI
5	ATC

dep_loc

<u>dep</u>	<u>loc</u>
1	Logroño
5	Arnedo

```
select *  
from dep cross join  
dep_loc
```

num	nomb	dep	loc
1	DMC	1	Logroño
1	DMC	5	Arnedo
4	LSI	1	Logroño
4	LSI	5	Arnedo
5	ATC	1	Logroño
5	ATC	5	Arnedo

JOIN

dep

<u>num</u>	nomb
1	DMC
4	LSI
5	ATC

dep_loc

<u>dep</u>	<u>loc</u>
1	Logroño
5	Arnedo

CE:dep


Con
condición
apropiada

```
select *  
from dep, dep_loc  
where num=dep
```

num	nomb	dep	loc
1	DMC	1	Logroño
5	ATC	5	Arnedo

Columnas de igual nombre /1


dep



<u>dep</u>	nomb
1	DMC
4	LSI
5	ATC



loc_dep

<u>loc</u>	<u>dep</u>
Logroño	1
Arnedo	5



Uso de * en
select para
una tabla


```
select dep.nomb, loc_dep.*  
from dep, loc_dep  
where dep.dep=loc_dep.dep
```



con nombres
de tabla

Columnas de igual nombre /2


dep





<u>dep</u>	nomb
1	DMC
4	LSI
5	ATC

loc_dep

<u>loc</u>	<u>dep</u>
Logroño	1
Arnedo	5



```
select D.nomb, L.*  
from dep as D, loc_dep as L  
where D.dep=L.dep
```



con alias
de tabla

El "as" en
el from es
opcional

Join explícito

```
select D.nomb, L.loc  
from dep as D, loc_dep as L  
where D.dep=L.dep
```

en
Access
inner join



```
select D.nomb, L.loc  
from dep as D join  
loc_dep as L on D.dep=L.dep
```

el join y su
condición
se indican
en from

Ejercicios con join

50. Nombres de asignatura de la titulación de Matemáticas
116. Nombres de asignatura y créditos donde está/n matriculada/s 'Rosa' 'García'



On: atributos de ambas tablas

```
select A.nom, P.DNI  
from asignatura as A join  
profesor as P on  
A.idprof=P.idprof  
and P.loc=Sevilla'  
where P.loc='Sevilla'
```

entre atributos,
uno de una tabla y
el otro de la otra
tabla

esta no es
condición
de join

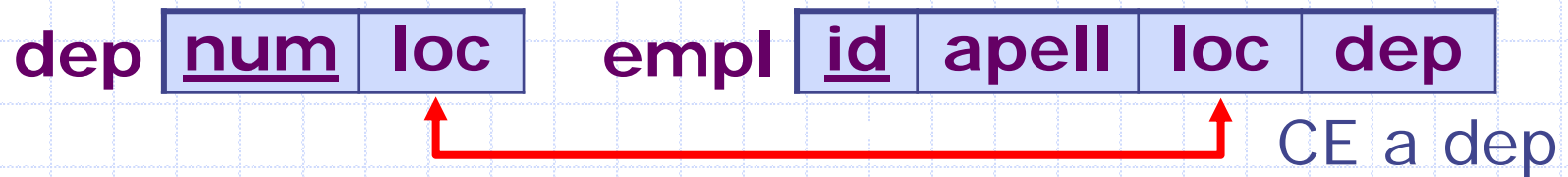
On: puede usarse <, >, <>...

```
select A.nom, P.DNI  
from asignatura as A join  
profesor as P on  
A.idprof>=P.idprof
```

Cualquier
comparación:

=, <>, >, <, >=, <=,
not, and, or

On: no sólo primarias/extranjeras



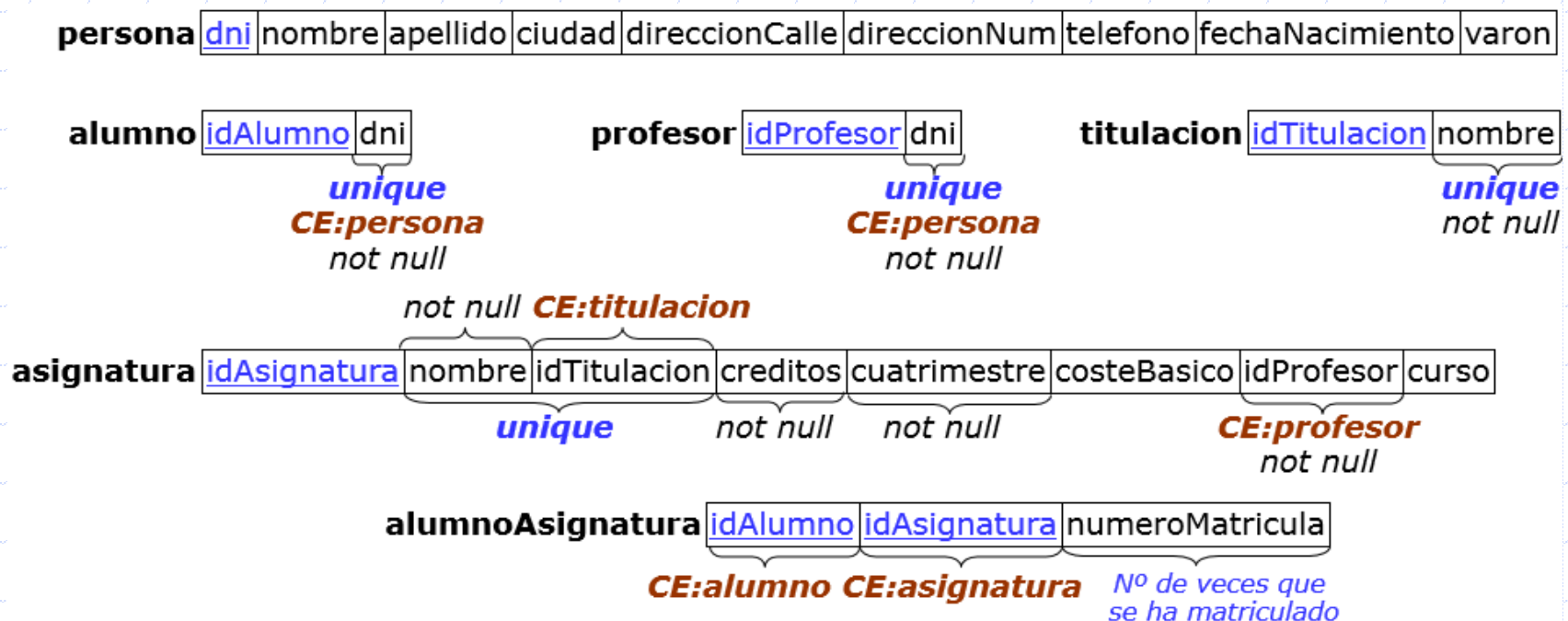
```
select D.num, E.apell  
from dep as D join  
empl as E on D.loc=E.loc
```

Los atributos loc
no son claves
primarias ni
extranjeras

Ejercicios con join

35. Dni de los profesores que figuran también como alumnos, junto con el nombre de las asignaturas en las que están matriculados.

Ejercicios 32, 33, 34, 37, 42



Repetición de tabla en join

ejemplo de
producto
cartesiano

```
select J.cod as J,  
       S.cod as S  
from empleado as J,  
     empleado as S
```

empleado

<u>cod</u>	nom	jefe
1	Miguel	2
2	Alicia	6
6	Lorena	null

cada aparición
de **empleado**
necesita su alias

J	S
1	1
1	2
1	6
2	1
2	2
2	6
6	1
6	2
6	6

Autojoin

Parejas de códigos de jefe y su subordinado directo

cada aparición
de empleado
necesita su alias

```
select J.nom as jefe,  
       S.nom as subord  
from empleado as J join  
     empleado as S on  
       J.cod=S.jefe
```

empleado

<u>cod</u>	nom	jefe
1	Miguel	2
2	Alicia	6
6	Lorena	null
7	Félix	8
8	Javier	6

jefe	subord
Alicia	Miguel
Lorena	Alicia
Félix	Javier
Javier	Lorena

Ejercicios con autojoin

45. DNI, nombre y apellido de los alumnos que tienen a 'Jorge' 'Sáenz' como profesor
51. Listado de asignaturas con más créditos que Seguridad Vial

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion credits cuatrimestre costeBasico idProfesor curso

not null CE:titulacion

unique *not null* *not null* *CE:profesor*
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Ejercicios con autojoin

66. Identificador de los alumnos matriculados en las asignaturas 1502 y 1301 (en ambas)
117. Identificador, nombre y titulación (nombre) de las diferentes asignaturas que tienen otra de igual nombre en otra titulación



Ejercicios con autojoin

118. Parejas de dni de alumno y profesor, donde ambos tienen el mismo nombre y apellido, exceptuando las que correspondan a la misma persona

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null *CE:titulacion* not null not null *CE:profesor* not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios con autojoin

119. DNI, nombre y apellido de los profesores con asignaturas en las titulaciones 7 y 10. Modificarlo después para que sean las titulaciones de matemáticas e informática.

persona	<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
----------------	------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

alumno	<u>idAlumno</u>	dni
---------------	-----------------	-----

unique
CE:persona
not null

profesor	<u>idProfesor</u>	dni
-----------------	-------------------	-----

unique
CE:persona
not null

titulacion	<u>idTitulacion</u>	nombre
-------------------	---------------------	--------

unique
not null

asignatura	<u>idAsignatura</u>	nombre	idTitulacion	creditos	cuatrimestre	costeBasico	idProfesor	curso
-------------------	---------------------	--------	--------------	----------	--------------	-------------	------------	-------

unique (nombre, idTitulacion)
not null (idTitulacion)
not null (creditos)
not null (cuatrimestre)
CE:profesor (idProfesor)
not null (idProfesor)

alumnoAsignatura	<u>idAlumno</u>	<u>idAsignatura</u>	numeroMatricula
-------------------------	-----------------	---------------------	-----------------

CE:alumno **CE:asignatura** Nº de veces que se ha matriculado

Join natural: condición implícita

dep

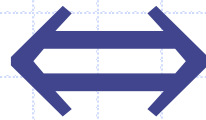
<u>num</u>	loc
------------	-----

 empl

<u>id</u>	apell	loc	dep
-----------	-------	-----	-----

CE a dep

```
select *  
from dep as D  
  natural join  
empl as E
```



```
select *  
from dep as D join  
empl as E  
  on D.loc=E.loc
```

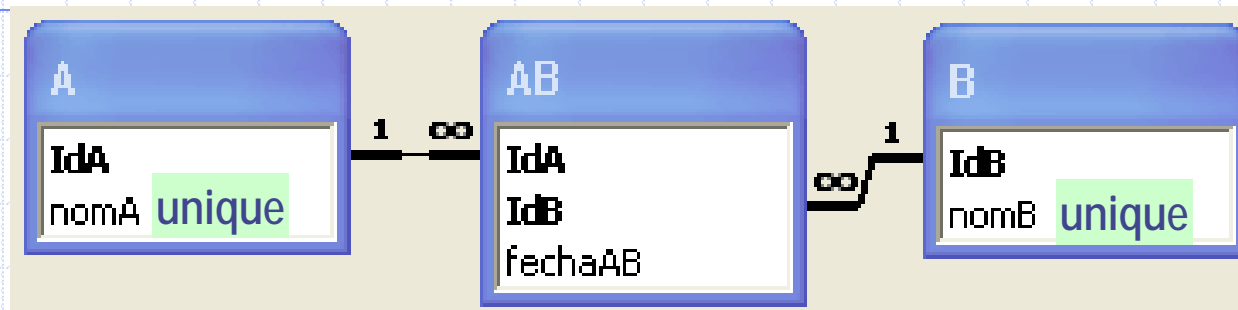
condición implícita:
igualdad de atributos

¿Alias de un join?

```
select *  
from (dep as D join  
      loc_dep as L  
      on D.num= L.dep) as DL join  
persona ...
```

El estándar SQL no lo admite, algunos sistemas sí

Revisando el join /1

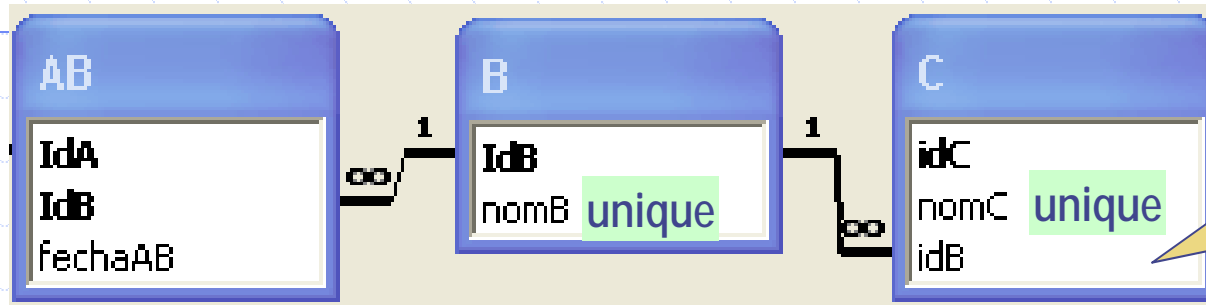


```
select A.nomA, B.nomB
from A join
      AB on A.idA=AB.idA join
      B  on AB.idB=B.idB
```

La tabla resultado tendrá una fila por cada fila de...

- a) La tabla A
- b) La unión de las tablas A y B
- c) La tabla AB
- d) No puedo saberlo

Revisando el join /2

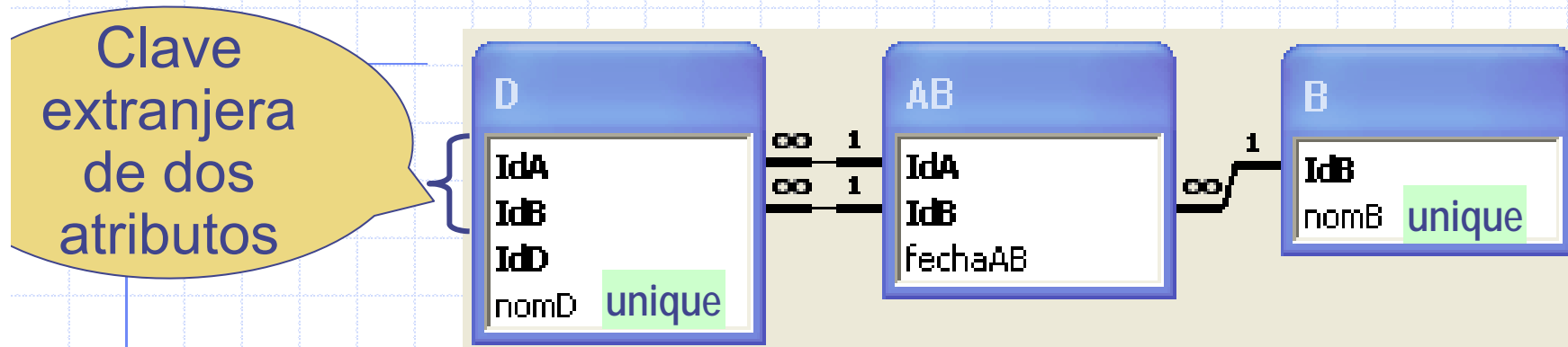


```
select AB.fechaAB, B.nomB , C.nomC
from AB join
      B   on AB.idB=B.idB join
      C   on B.idB=C.idB
```

La tabla resultado tendrá una fila por cada fila de...

- a) La tabla AB
- b) La unión de AB y C
- c) La unión de B y C
- d) No puedo saberlo

Revisando el join /3



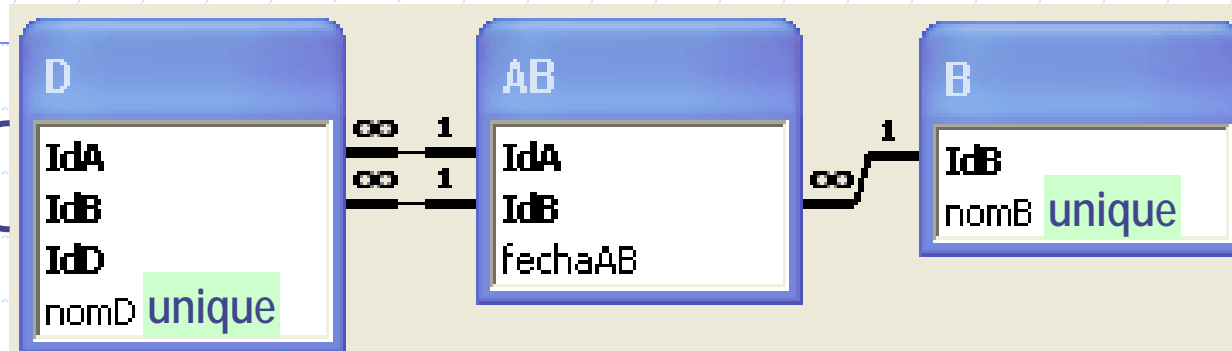
```
select D.nomD, AB.fechaAB, B.nomB
from D join
AB on (D.idA,D.idB) = (AB.idA,AB.idB) join
B on AB.idB=B.idB
```

La tabla resultado tendrá una fila por cada fila de...

- a) La tabla D
- b) La tabla AB
- c) La unión de B y D
- d) No puedo saberlo

Revisando el join /4

Mismo
esquema
de antes



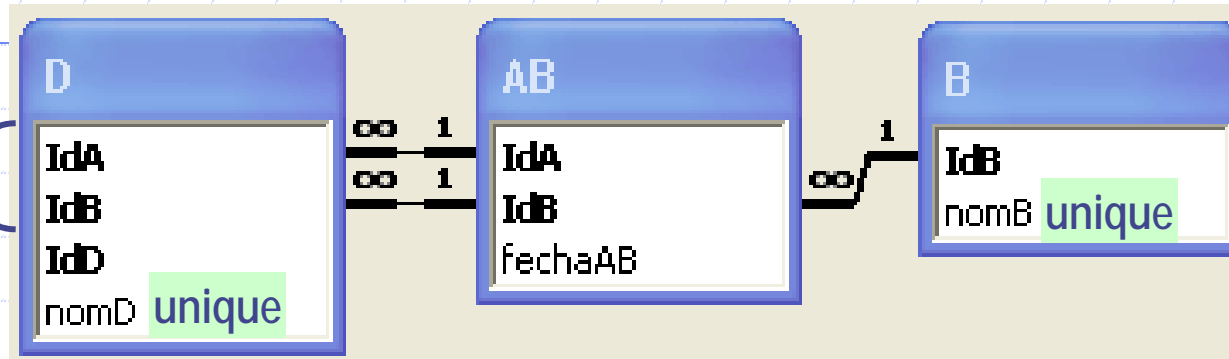
```
select  D.nomD, AB.fechaAB, B.nomB
from    D join
        AB on D.idB=AB.idB join
        B  on AB.idB=B.idB
```

La tabla que por cada fila produce una fila resultado es...

- a) La tabla D
- b) La tabla AB
- c) La unión de B y D
- d) No puedo saberlo

Revisando el join /5

Mismo
esquema
de antes



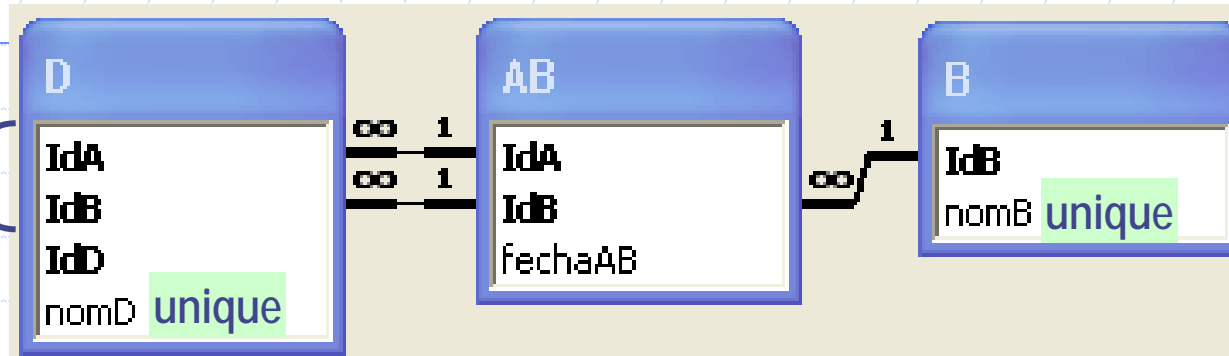
```
select D.nomD, AB.fechaAB, B.nomB
from (D natural join
      AB) natural join
      B
```

Esta consulta produce

- a) El mismo resultado anterior
- b) Error en el segundo join
- c) Error ya que SQL no tiene este join
- d) Error porque B no tiene IdA

Revisando el join /6

Mismo
esquema
de antes

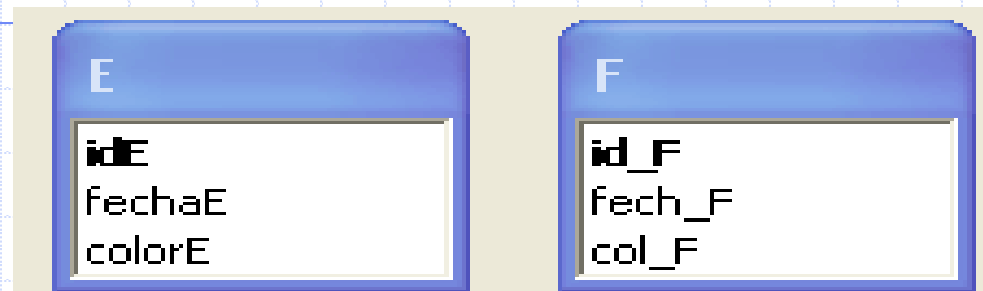


```
select D.nomD, B.nomB
from D join
      AB on (D.idA,D.idB) = (AB.idA,AB.idB) join
      B on AB.idB=B.idB
```

En esta consulta

- a) Se pueden producir filas repetidas
- b) Sobra la tabla AB
- c) El on entre D y AB debería tener sólo idB
- d) Obtenemos B cross join D

Revisando el join /7



```
select E.idE
from E join
      F on E.fechaE = F.fech_F and
           E.colorE = F.col_F
```

El join anterior siempre debería comparar entre:

- a) Una clave primaria con una extranjera
- b) Claves primarias, entre extranjeras o entre prim/extr
- c) Atributo/s de E con atributo/s de F de tipo compatible
- d) Teóricamente cualquier cosa, incluso `colorE='azul'`

Ejercicios con join

120. Nombre de los empleados que trabajan en proyectos de la misma localidad donde residen (la localidad del empleado es la parte final de la dirección, tras coma y espacio).

empleado NSS nombre apellidos fechaNcto direccion sexo salario jefe depto

not null *not null* **CE:empleado** **CE:departamento**

proyecto numero nombre localidad depto
unique CE:departamento

trabaja empleado proyecto horas

CE:empleado CE:proyecto

departamento	<u>numero</u>	nombre	director	fechaIniDir
		<i>unique</i>	<i>CE:empleado</i>	
		<i>not null</i>		

Localidades **Depto** depto localidad
CE: departamento

familiar empleado nombre fechaNcto sexo parentesco

CE: empleado

Ejercicios con join

121. Código sin repeticiones de los libros con dos o más autores



Índice / consultas

1. Sobre una tabla
2. Join
- 3. Group by y having**

Group by

No confundirlo con **order by**

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

```
select loc, count(*)  
from persona  
group by loc
```

loc	count(*)
Bilbao	1
Haro	2
Logroño	3
Madrid	2

Una fila por cada **valor de loc**

Group by: ¿Qué pongo en select?

¡Esta mal!
Aunque es posible
que el gestor no de
error

persona

```
select apell, count(*)  
from persona  
group by loc
```

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

apell	count(*)
???	1
???	2
???	3
???	2

diferentes apell
para el mismo
valor de loc

Group by: ¿en select sólo funciones?

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

Posible

```
select count(*),  
       min(apell)  
from persona  
group by loc
```

count(*)	min(apell)
1	Sánchez
2	Díaz
3	Díaz
2	Beltrán

Group by con varias columnas

Incluso se pueden poner expresiones

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

```
select loc,  
       year(fNac) año,  
       count(*) num  
from persona  
group by loc,  
         year(fNac)
```

loc	año	num
Bilbao	66	1
Haro	78	2
Logroño	71	2
Logroño	78	1
Madrid	74	1
Madrid	80	1

e SQL

Group by: ¿en select sólo parte?

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

Posible

```
select year(fNac) año,  
count(*) num  
from persona  
group by year(fNac),  
loc
```

año	num
66	1
78	2
71	2
78	1
74	1
80	1

Resultado anterior diferente a este

persona

ya que
cambia el
criterio de
los grupos

```
select year(fNac) año,  
count(*) num  
from persona  
group by year(fNac)  
,  
loc
```

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

año	num
66	1
78	2
71	2
78	1, 3
74	1
80	1

Group by: también sin funciones

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

Posible

```
select year(fNac) año,  
loc  
from persona  
group by year(fNac),  
loc
```

año	loc
66	Bilbao
78	Haro
71	Logroño
78	Logroño
74	Madrid
80	Madrid

Máximo 1 nivel de agrupación

Un nivel de agrupación

```
select loc,  
       apell  
       count(*) num  
from persona  
group by loc,  
       apell
```

No permite group by
de otra agrupación
previa

```
select loc,  
       apell  
       count(*) num  
from persona  
group by loc (  
group by apell, nom)
```

Ejercicios con group by

122. Nombre de titulación, curso y cuatrimestre con a su nº de asignaturas y coste básico medio.

55. Nombre de cada asignatura y número de alumnos matriculados en ella.

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

not null CE:titulacion

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

unique

not null

not null

CE:profesor
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura

Nº de veces que se ha matriculado

Ejercicios con group by

123. Cada nueva matrícula de una asignatura (numeroMatricula) supone un 10% de aumento sobre el coste básico. Mostrar el identificador de cada alumno junto a lo que ha pagado en concepto de matrícula.

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null not null **CE:profesor** not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios con group by

124. Nombre de biblioteca, n° de libros distintos prestados y n° de prestamos hechos (de las bibliotecas que hayan hecho algún préstamo)

libro	<u>id</u>	titulo	editorial	año	peso	paginas	precio	primer_solicitante
--------------	-----------	--------	-----------	-----	------	---------	--------	--------------------

CE:editorial

editorial	<u>nombre</u>	direccion	tfno
------------------	---------------	-----------	------

CE:lector

autor	<u>id</u>	nombre	apellidos
--------------	-----------	--------	-----------

autorlibro	<u>autor</u>	<u>libro</u>
-------------------	--------------	--------------

CE:autor CE:libro

lector	<u>id</u>	nombre	apellidos	direccion	tfno	fecha_inscripcion
---------------	-----------	--------	-----------	-----------	------	-------------------

biblioteca	<u>id</u>	nombre	direccion	tfno
-------------------	-----------	--------	-----------	------

ejemplar	<u>libro</u>	<u>ejemplar</u>	biblioteca	fecha_solicitud	fecha_compra
-----------------	--------------	-----------------	------------	-----------------	--------------

CE:libro

CE:biblioteca

not null

prestamo	<u>libro</u>	<u>ejemplar</u>	<u>fecha_prestado</u>	fecha_devuelto	lector
-----------------	--------------	-----------------	-----------------------	----------------	--------

CE:ejemplar

CE:lector

not null

Group by y null

¡ grupo con
el valor null !

```
select depto,  
       count(*) as n,  
       avg(salario) as s  
from empleado  
group by depto
```

empleado

nom	salario	depto
Juan	null	5
Felisa	10.000	null
Ramón	20.000	null
Abel	10.000	4
Iván	20.000	1

depto	n	s
1	1	20.000
4	1	10.000
5	1	null
null	2	15.000

Having: condición de grupo

Posible: funciones
agrupadas como sum,
min... (en where no)

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

```
select loc, count(*)  
from persona  
group by loc  
having min(fNac)  
    >=1/1/72
```

loc	count(*)
Bilbao	1
Haro	2
Logroño	3
Madrid	2

Having: puede haber del group by

persona

Posible:
condiciones
con atributos
del group by

```
select loc, count(*)  
from persona  
group by loc  
having loc like '%d'
```

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

loc	count(*)
Bilbao	1
Haro	2
Logroño	3
Madrid	2

Having condiciones posibles

having es una
condición para
el grupo

persona

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

```
select loc, count(*)  
from persona  
group by loc  
having year(fNac)  
=72
```

¿cuál fecha
de las del
grupo?

da
ERROR

Having: funciones + atrib. group

persona

Posible:
mezcla de
funciones y
atributos del
group by

nom	apell	loc	fNac
Luis	Díaz	Haro	1/1/78
Laura	Beltrán	Madrid	8/8/74
María	Lasa	Logroño	1/10/71
Pepe	García	Madrid	2/2/80
Juan	Sánchez	Bilbao	3/3/66
Rosa	Díaz	Logroño	11/1/71
Rosa	García	Haro	4/4/78
Jorge	Sánchez	Logroño	9/9/78

```
select loc, count(*)  
from persona  
group by loc  
having loc like '%o'  
and count(*)>1
```

loc	count(*)
Bilbao	1
Haro	2
Logroño	3
Madrid	2

Group by/having orden ejecución

Localidad y nº de nacidos después de 1971
para localidades sin nacidos después del 78

persona

apell	loc	fNac
Díaz	Haro	1/1/78
Beltrán	Madrid	8/8/74
Lasa	Logroño	1/10/71
García	Madrid	2/2/80
Sánchez	Bilbao	3/3/66
Díaz	Logroño	11/1/71
García	Haro	4/4/78
Sánchez	Logroño	9/9/78

```
4 select loc, count(*)  
  from persona  
1 where year(fNac)>71  
2 group by loc  
3 having max(fNac)<1/1/79
```

Primero se hace where

Localidad y nº de nacidos después de 1971
para localidades sin nacidos después del 78

apell	loc	fNac
Díaz	Haro	1/1/78
Beltrán	Madrid	8/8/74
Lasa	Logroño	1/10/71
García	Madrid	2/2/80
Sánchez	Bilbao	3/3/66
Díaz	Logroño	11/1/71
García	Haro	4/4/78
Sánchez	Logroño	9/9/78

1

```
select loc, count(*)  
from persona  
where year(fNac)>71  
group by loc  
having max(fNac)<1/1/79
```

podemos pensar
que primero se
harían los join

Segundo se agrupa

Localidad y n° de nacidos después de 1971
para localidades sin nacidos después del 78

apell	loc	fNac
Díaz	Haro	1/1/78
García	Haro	4/4/78
Sánchez	Logroño	9/9/78
García	Madrid	2/2/80
Beltrán	Madrid	8/8/74

2

```
select loc, count(*)  
from persona  
where year(fNac)>71  
group by loc  
having max(fNac)<1/1/79
```

Tercero se hace having

Localidad y nº de nacidos después de 1971
para localidades sin nacidos después del 78

apell	loc	fNac
Díaz	Haro	1/1/78
García	Haro	4/4/78
Sánchez	Logroño	9/9/78
García	Madrid	2/2/80
Beltrán	Madrid	8/8/74

3

```
select loc, count(*)  
from persona  
where year(fNac)>71  
group by loc  
having max(fNac)<1/1/79
```


Por último se calcula select

Localidad y nº de nacidos después de 1971
para localidades sin nacidos después del 78

apell	loc	fNac
Díaz	Haro	1/1/78
García	Haro	4/4/78
Sánchez	Logroño	9/9/78

4 `select loc, count(*)
from persona
where year(fNac)>71
group by loc
having max(fNac)<1/1/79`

loc	count(*)
Haro	2
Logroño	1

Ejercicios group by /having

125. Código y nombre de titulación, nº de asignaturas y media del coste básico de las mismas, para titulaciones cuyo coste básico total de las asignaturas sea mayor de 600€

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

unique

not null

not null

CE:profesor
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Ejercicios group by /having

61. Apellido y nombre de los alumnos con dos asignaturas o más, ordenados alfabéticamente.
62. Apellido y nombre de alumnos que pagaron más de 500€ por asignaturas de la titulación de 'Matemáticas'

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

not null CE:titulacion

unique

not null

not null

CE:profesor
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Ejercicios group by /having

66. Identificador de los alumnos matriculados en las asignaturas 150212 y 130113 (en ambas a la vez).

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
not null CE:titulacion
unique *not null* *not null* *CE:profesor*
 not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Ejercicios group by/having

126. Código de los clientes que han gastado en su ciudad un total de más de 20 €
127. Código de tienda y nº de clientes procedentes de otra localidad (diferentes) a los que la tienda les ha vendido algo (si no ha vendido nada a gente de otras localidades no sacarla).

cliente	<u>ident_client</u>	apell1	nombr_client	nacion	ciud
	not null				

tienda	<u>ident_tienda</u>	ciud	nombr_gerent
	not null		

proveedor	<u>ident_prov</u>	nombr_prov
	not null	

articulo	<u>ident_artic</u>	nombr_artic	peso	color	prec_compr	prec_vent	ident_prov
	not null				not null	not null	CE:proveedor

venta	<u>ident_client</u>	<u>ident_tienda</u>	<u>ident_artic</u>	<u>fech_vent</u>	cant_artics	prec_total
	CE:cliente	CE:tienda	CE:articulo			

Ejercicios group by/having

118'. Parejas de dni de alumno y profesor, donde ambos tienen el mismo nombre y apellido, exceptuando las que correspondan a la misma persona ...para alumnos con más de 5 asignaturas y profesores de Logroño

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

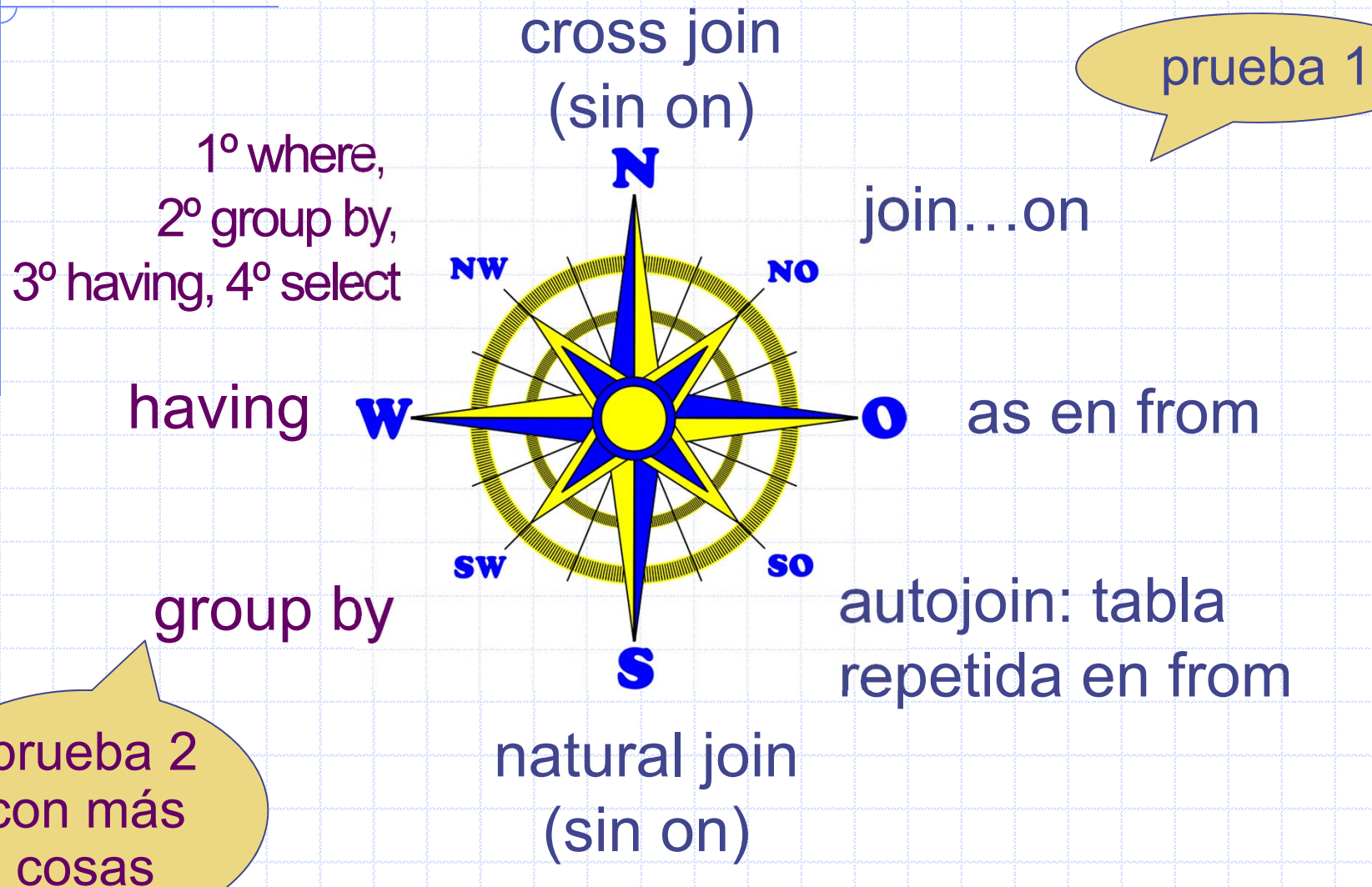
profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion credits cuatrimestre costeBasico idProfesor curso
unique not null not null **CE:profesor**
 not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Varias tablas y agrupación



Índice

- ◆ Introducción
- ◆ LMD: consultas
- ◆ **LDD**
- ◆ LMD: actualizaciones
- ◆ LDD
- ◆ LMD: más consultas

Crear vistas

Una vista es **una consulta** almacenada en la BD con un **nombre**

Nombre y apellido de los empleados y nº de horas por semana que dedica a cada proyecto, indicando el nombre del proyecto.

en Access se llaman consultas

Create view **EmplConProy** as

```
select E.nom, E.apellido, T.horas, P.nom as pry
from empleado E join
trabaja T on E.NSS=T.Empleado join
proyecto P on T.proyecto=P.numero;
```

Uso de vistas en consultas

Se utiliza usando el nombre que le hemos puesto **como si fuera otra tabla**:


Nombre y apellido de los empleados que trabajan en el proyecto de nombre 'ProductoX'

```
select nom, apellido  
from EmplConProy  
where pry='ProductoX';
```

Nombre de
la vista
creada antes

Funcionamiento de una vista

- ◆ Una vista es una **tabla virtual**
- ◆ Al consultarla se **recalculan** sus filas
- ◆ Pueden **materializarse**, es decir que su resultado se guarde “congelado” en la BD



Las vistas
materializadas
no se ven este
curso

Borrado de vistas

Drop view EmplConProy;

Ejercicios create view

201. Crea la vista **PresNoDevuelto** que obtenga los préstamos no devueltos hechos hace 365 días o más. Escribir después una consulta con esa vista que obtenga código, título de libro y nº de préstamos no devueltos hechos hace 365 días o más.

libro	<u>id</u>	titulo	editorial	año	peso	paginas	precio	primer_solicitante
	<i>CE:editorial</i>						<i>CE:lector</i>	

editorial	<u>nombre</u>	direccion	tfno
------------------	---------------	-----------	------

autor	<u>id</u>	nombre	apellidos
--------------	-----------	--------	-----------

autorlibro	<u>autor</u>	<u>libro</u>
	<i>CE:autor</i>	<i>CE:libro</i>

lector	<u>id</u>	nombre	apellidos	direccion	tfno	fecha_inscripcion
---------------	-----------	--------	-----------	-----------	------	-------------------

biblioteca	<u>id</u>	nombre	direccion	tfno
-------------------	-----------	--------	-----------	------

ejemplar	<u>libro</u>	<u>ejemplar</u>	biblioteca	fecha_solicitud	fecha_compra
	<i>CE:libro</i>		<i>CE:biblioteca</i>		
			<i>not null</i>		

prestamo	<u>libro</u>	<u>ejemplar</u>	<u>fecha_prestado</u>	fecha_devuelto	lector
	<i>CE:ejemplar</i>				<i>CE:lector</i>
					<i>not null</i>

Índice

- ◆ Introducción
- ◆ LMD: consultas
- ◆ LDV
- ◆ **LMD: actualizaciones**
- ◆ LDD
- ◆ LMD: más consultas

Start transaction / commit

- ◆ **Start transaction** inicia una *transacción* (conjunto de operaciones)
- ◆ **Commit** finaliza la transacción fijando en la BD los cambios ocurridos desde start transaction
- ◆ **Se usan en algunos laboratorios**

Start transaction / rollback

- ◆ **Rollback** deshace los cambios ocurridos desde el start transaction anterior
- ◆ **Rollback** no puede deshacer cambios de borrado de tablas (drop table)
- ◆ **Se usan en algunos laboratorios**

Insert: introduce una fila

titulacion idTitulacion nombre
unique
not null

```
insert into titulacion  
values ('170100', 'Magisterio')
```

Insert: orden de los valores

titulacion idTitulacion nombre
unique
not null

~~insert into titulacion
values ('Arquitectura', '180800')~~

Orden
inadecuado
de valores

Insert: se pueden usar nulos

alumnoAsignatura

<u>idAlumno</u>	<u>idAsignatura</u>	numeroMatricula
-----------------	---------------------	-----------------

CE:alumno CE:asignatura

*Nº de veces que
se ha matriculado*

```
insert into alumnoAsignatura  
values ('A131313', '130122', null)
```

Se pone sin
comillas

Insert: con nombres de atributo

titulacion idTitulacion nombre
unique
not null

```
insert into titulacion (Nombre, IdTitulacion)  
values ('Arquitectura','180800')
```

orden
apropiado entre
atributos y
valores

Insert: sólo algunos atributos

titulacion idTitulacion nombre
unique

```
insert into titulacion (idTitulacion)  
values ('190000')
```

El resto de atributos
tomará el valor por defecto
(no siempre es nulo)

Insert: con consulta dentro

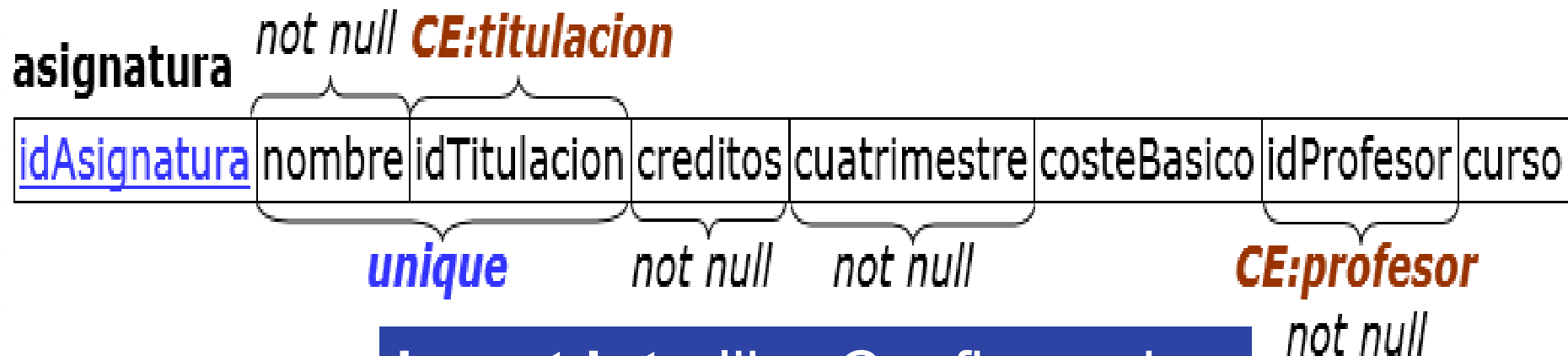
titulacion idTitulacion nombre
unique
not null

```
insert into titulación  
values('150000', (select nombre  
from master  
where id='123'))
```

Las
consultas
van entre
paréntesis

En este caso la
consulta debe
devolver una fila y
una columna

Insert: varias filas con consulta

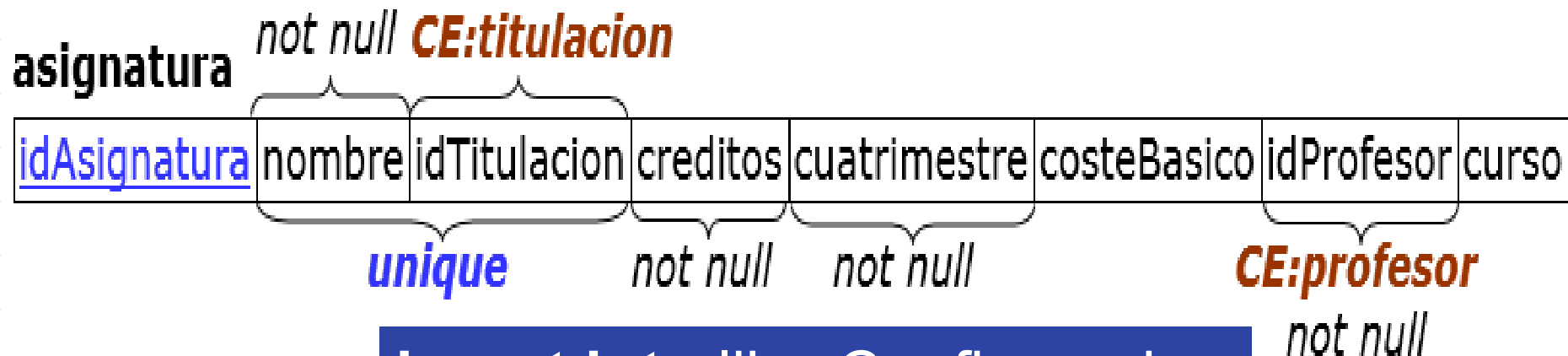


```
insert into libreConfiguracion
select *
from asignatura
where IdTitulacion is null
```

Se insertan tantas
filas como devuelva
la consulta

libreConfiguracion
se ha definido igual
que asignatura

Insert: varias filas con consulta



```
insert into libreConfiguracion
select *
from asignatura
where IdTitulacion is null
```

También se puede
usar con atributos
entre paréntesis

Ejercicios insert

202. Introducir el libro ("L111", "Bases de Datos", "Editores Reunidos", 2015, 34, 160, 33, "Le48"). El autor es Andrés Pérez de código A12. Ni este autor ni la editorial (dirección La Paz 5, Logroño) están en la BD, pero sí el 1º solicitante.

libro	<u>id</u>	titulo	editorial	año	peso	paginas	precio	primer_solicitante
			CE:editorial					CE:lector

editorial	<u>nombre</u>	direccion	tfno
------------------	---------------	-----------	------

autor	<u>id</u>	nombre	apellidos
--------------	-----------	--------	-----------

autorlibro	<u>autor</u>	<u>libro</u>
-------------------	--------------	--------------

CE:autor CE:libro

lector	<u>id</u>	nombre	apellidos	direccion	tfno	fecha_inscripcion
---------------	-----------	--------	-----------	-----------	------	-------------------

biblioteca	<u>id</u>	nombre	direccion	tfno
-------------------	-----------	--------	-----------	------

ejemplar	<u>libro</u>	<u>ejemplar</u>	biblioteca	fecha_solicitud	fecha_compra
-----------------	--------------	-----------------	------------	-----------------	--------------

CE:libro

CE:biblioteca

not null

prestamo	<u>libro</u>	<u>ejemplar</u>	<u>fecha_prestado</u>	fecha_devuelto	lector
-----------------	--------------	-----------------	-----------------------	----------------	--------

CE:ejemplar

CE:lector

not null

Ejercicios insert

203. Incluir en la biblioteca **P1** el ejemplar de código de ejemplar **L112** del único libro del autor **Pepe Pérez**. La fecha de solicitud es la de hoy y la de compra nula.

libro id titulo editorial año peso paginas precio primer_solicitante
CE:editorial *CE:lector*

editorial nombre direccion tfno

autor id nombre apellidos

autorlibro autor libro
CE:autor CE:libro

lector id nombre apellidos direccion tfno fecha_inscripcion

biblioteca id nombre direccion tfno

ejemplar libro ejemplar biblioteca fecha_solicitud fecha_compra
CE:libro CE:biblioteca
 not null

prestamo libro ejemplar fecha_prestado fecha_devuelto lector
CE:ejemplar CE:lector
 not null

Update: modifica filas

persona

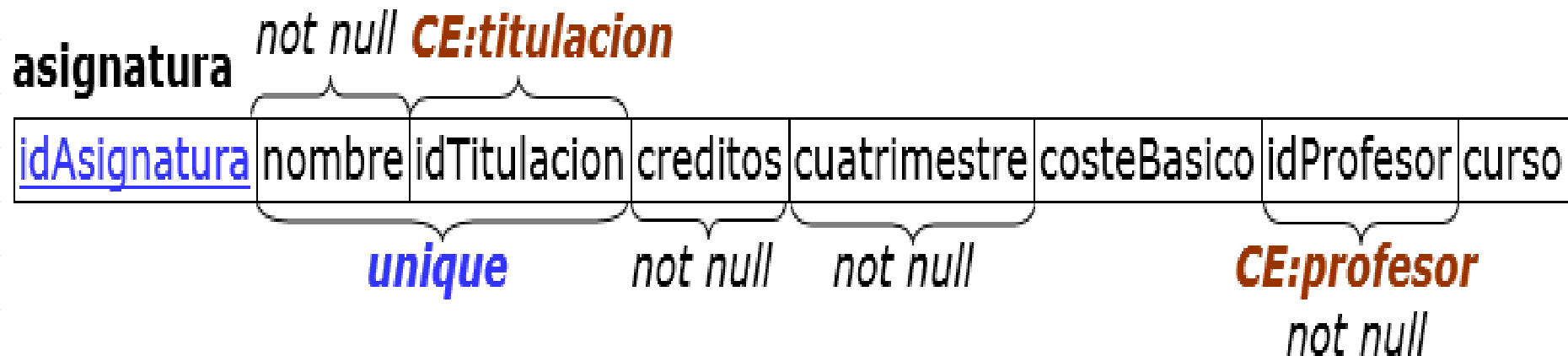
<u>dni</u>	nombre	apellido	ciudad	direccionCalle	direccionNum	telefono	fechaNacimiento	varon
------------	--------	----------	--------	----------------	--------------	----------	-----------------	-------

```
update persona
set    telefono='941111112'
where nombre='Luis' and apellido='Ramírez'
```

Realiza las
modificaciones
de set

Modifica las filas
que cumplen la
condición

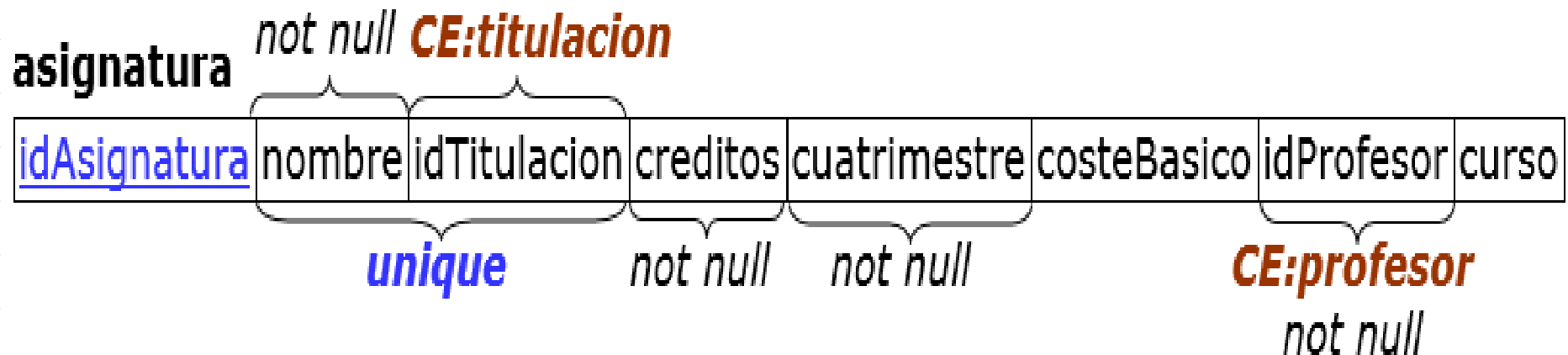
Update: varias modificaciones



```
update asignatura
set    creditos=creditos-1,
       costeBasico=costeBasico*1.1
where  IdTitulación='13110'
```

Modifica
varias
columnas

Update: consultas en set o where



```
update asignatura
set costeBasico=costeBasico*0,9
where IdAsignatura in
(select IdAsignatura
from alumnoAsignatura
group by IdAsignatura
having count(*)>100
)
```

entre
paréntesis

in es €
(lo vimos
en
consultas
básicas)

Update: consultas en set o where

Se puede usar **in** y **not in**

Se pueden usar **=**, **<>**, **<**, **>**, **<=**, **>=** si la consulta devuelve sólo un dato

```
update asignatura
set   costeBasico=costeBasico*0,9
where IdAsignatura <>
(select IdAsignatura
 from alumnoAsignatura
 ....)
```

la consulta
debe devolver
1 fila y 1
columna

Update: correlación con consultas

```
update asignatura A
set    A.costeBasico=A.costeBasico*0,9
where  (select count(*)
        from alumnoAsignatura AA
        where AA.IdAsignatura =
              A.IdAsignatura -- correlación
       ) > 100
```

¡ también hay
correlaciones en
insert y delete !

Delete: borra filas

alumnoAsignatura

<u>idAlumno</u>	<u>idAsignatura</u>	numeroMatricula
-----------------	---------------------	-----------------

CE:alumno CE:asignatura

*Nº de veces que
se ha matriculado*

```
delete from alumnoAsignatura  
where IdAlumno='A131313'
```

Borra las filas
que cumplen la
condición

Delete: ¿borrar todas las filas?

alumnoAsignatura

<u>idAlumno</u>	<u>idAsignatura</u>	numeroMatricula
-----------------	---------------------	-----------------

CE:alumno CE:asignatura

*Nº de veces que
se ha matriculado*

delete from alumnoAsignatura

Sin condición
borra todas
las filas

Delete: where con consultas

alumnoAsignatura

<u>idAlumno</u>	<u>idAsignatura</u>	numeroMatricula
-----------------	---------------------	-----------------

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

```
delete from alumnoAsignatura
where IdAlumno =
  (select a.IdAlumno
   from alumno a join persona p
   on a.dni=p.dni
   where p.nombre='Rosa' and
        p.apellido='García')
```

también
admite
correlaciones

Ejercicios delete/update

204. Se va a cerrar la biblioteca de **Almendralejo**. Se van a trasladar sus ejemplares a la biblioteca de **Haro**. Hay que eliminar la biblioteca y la información de sus préstamos.

libro id titulo editorial año peso paginas precio primer_solicitante
CE:editorial *CE:lector*

editorial nombre direccion tfno

autor id nombre apellidos

autorlibro autor libro

CE:autor CE:libro

lector id nombre apellidos direccion tfno fecha_inscripcion

biblioteca id nombre direccion tfno

ejemplar libro ejemplar biblioteca fecha_solicitud fecha_compra

CE:libro

CE:biblioteca

not null

prestamo libro ejemplar fecha_prestado fecha_devuelto lector

CE:ejemplar

CE:lector

not null

Ejercicios delete/update

205. El lector **Mario Redondo** (sólo hay uno con ese nombre) se da de baja y hay que eliminarlo junto a toda la información de su actividad (préstamos y libros solicitados).

libro id titulo editorial año peso paginas precio primer_solicitante
CE:editorial *CE:lector*

editorial nombre direccion tfno

autor id nombre apellidos

autorlibro autor libro
CE:autor CE:libro

lector id nombre apellidos direccion tfno fecha_inscripcion

biblioteca id nombre direccion tfno

ejemplar libro ejemplar biblioteca fecha_solicitud fecha_compra
CE:libro CE:biblioteca
 not null

prestamo libro ejemplar fecha_prestado fecha_devuelto lector
CE:ejemplar CE:lector
 not null

Actualizar la BD

insert into tabla
values (...)

insert into tabla (...)
values (...)

consultas en values

insert into tabla ...
consulta

update tabla set col=...
where ...



update tabla
set col1=..., col2=..., ...
where col in (consulta)

update tabla **T**
set ...
where col >/in (consulta)

delete from tabla
where ...

delete from tabla
where col >/in (consulta)

correlaciones en
insert, delete y update

Índice

- ◆ Introducción
- ◆ LMD: consultas
- ◆ LDV
- ◆ LMD: actualizaciones
- ◆ **LDD**
- ◆ LMD: más consultas

Create table: definir una tabla

create table Departamento

Nombre
Tabla

(Nombre varchar(15),

Número int,

Jefe char (9),

FechaInicJefe date **default** '1-1-1',

Nombre
Columna

Tipo de
datos

Valor por
defecto

);

Tipos de datos

◆ Varían entre sistemas


◆ Básicos de MySQL:

- **char(n)**: string de longitud fija
- **varchar(n)**: string de longitud variable
- **int**: número entero
- **numeric(p,s)**: total **p** cifras y de ellas **s** decimales
- **date**: día, mes, año, horas, min. y seg.
Incluye varios formatos

Se autocompleta con espacios

Create table: restricción sin nulos

```
create table Departamento  
( Nombre    varchar(15) not null,  
  Número    int,  
  Jefe       char (9) not null,  
  FechaInicJefe date default '1-1-1',  
);
```



sin nulos

Create table: restricciones de clave

```
create table Departamento  
( Nombre varchar(15) not null,  
  Número int,  
  Jefe char (9) not null,  
  FechaInicJefe date default '1-1-1',  
  primary key(Número),  
  unique(Nombre),  
);
```

Primaria y
candidata

Create table: clave extranjera

```
create table Departamento  
( Nombre varchar(15) not null,  
  Número int,  
  Jefe char (9) not null,  
  FechaInicJefe date default '1-1-1',  
  primary key(Número),  
  unique(Nombre),  
  foreign key(Jefe) references Empleado(NSS)  
);
```

clave
extranjera

tabla(columnas)

Foreign key: on delete

`foreign key(Jefe) references Empleado(NSS)`

`on delete set null,`

Qué hacer si se va a **borrar** un valor de NSS y aparece en la clave extranjera Jefe

Foreign key: on update

foreign key(Jefe) references Empleado(NSS)
on update cascade
on delete set null,

Qué hacer si se va a **modificar** un valor de NSS y aparece en la clave extranjera Jefe

Acciones on update/delete

- ◆ **Set null:** null en clave extranjera
- ◆ **Set default:** valor por defecto en clave extranjera
- ◆ **Cascade:** {borra/modifica igualmente} las filas cuyo valor de clave extranjera es el valor de clave primaria a borrar/modificar
- ◆ **Restrict:** impide el borrado o la modificación
- ◆ Se hace **restrict** si no hay on delete/update

Create table: restricciones con nombre

```
create table Departamento
( Nombre varchar(15),
  Número int not null,
  Jefe char(9) not null,
  FechaInicJefe date default '1-1-1',
  constraint no_null_nom not null(Nombre),
  constraint pk_dpto primary key(Número),
  constraint cl_nomb unique(Nombre),
  constraint fk_jefe foreign key(Jefe)
                    references Empleado(NSS)
);
```

Ejercicios create table

206. Escribe las instrucciones de creación de las tablas de la BD Empresa en el orden adecuado

empleado NSS nombre apellidos fechaNcto direccion sexo salario jefe depto

not null not null **CE:empleadoCE:departamento**

proyecto

<u>numero</u>	nombre	localidad	depto
---------------	--------	-----------	-------

unique *CE:departamento*

trabaja empleado proyecto horas

CE:empleado CE:proyecto

departamento	<u>numero</u>	nombre	director	fechaIniDir
		<i>unique</i>	<i>CE:empleado</i>	<i>not null</i>

Localidades **Depto** depto localidad
CE: departamento

familiar empleado nombre fechaNcto sexo parentesco

CE: empleado

Create table: restricciones check

```
create table Asignatura
( Id          varchar(10) not null,
  Nombre      varchar(30) not null,
  Creditos    int          not null,
  Cuatrim     char(1)      not null
                                check(Cuatrim in ('1','2')),
  constraint cred_min check(Creditos>=3),
  constraint cred_max check(Creditos<=9),
  ...
);
```

también junto en una
constraint con
**check(creditos>=3
and creditos<=9)**

Create table: check entre campos

```
create table Ingresos
( Id          varchar(5)  not null,
  Concepto    varchar(40) not null
  ImporteMax  int,
  Importe     int,
  constraint imp_max check(Importe < ImporteMax),
  ...
);
```

NO se pueden
referenciar campos
de otras tablas

Hay que hacerlo en
cláusula constraint

Crear tabla mediante consulta

Crea las mismas
filas y columnas
que el resultado
de la consulta

opcional

```
create table personaAlumno (dni,nom,apell) as  
select P.dni, P.nombre, P.apellido  
from persona P join  
alumno A on P.dni=A.dni;
```

No se crea ninguna
restricción (claves,
sin nulos...)

Ejercicios create table

207. Crea la tabla **profesorPrimerC** con los profesores del 1º cuatrimestre. Llamar a las columnas idProf, dniProf, idAsig.

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null not null *CE:profesor* not null
 not null *CE:titulacion*

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios create table

208. Crea la tabla AsigMuchaMatricula con las asignaturas con más de 100 alumnos. Tendrá las mismas columnas que la tabla Asignatura.
209. Haz algo similar para asignaturas sin matrícula.

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null not null *CE:profesor* not null
 not null *CE:titulacion*

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Drop table: borrar la definición

Borra el **contenido** (filas) y también la **definición** de la tabla de la BD (esquema de la relación)

drop table departamento

No confundirlo con **delete from departamento** que borra sólo las filas

Drop table...cascade

Si hay referencias a la tabla (como claves extranjeras) desde otra lo anterior no funcionará, pero sí esto:

drop table departamento cascade

También borrará las referencias a la tabla departamento que haya en otras tablas

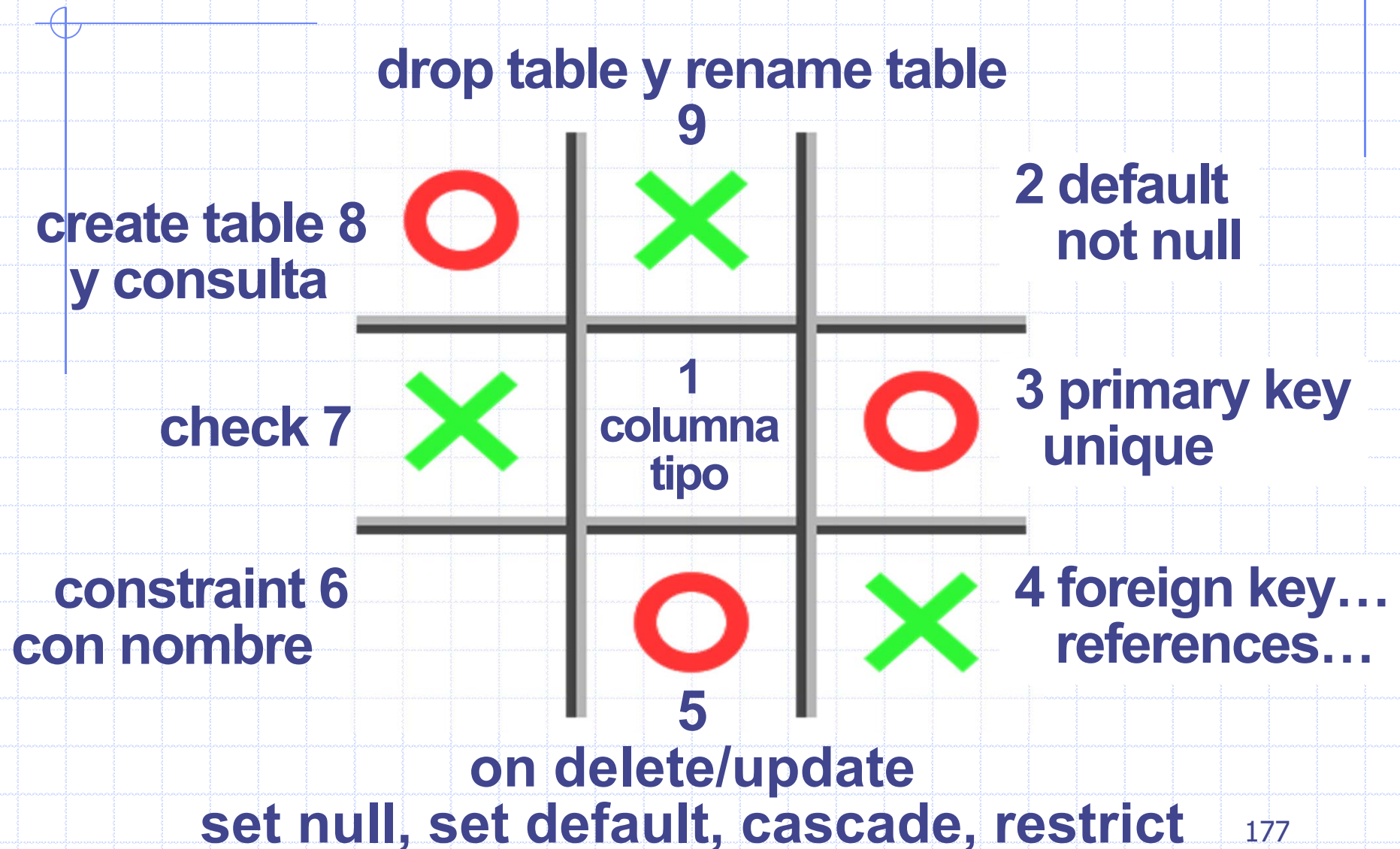
MySQL no admite cascade

Rename: cambiar nombre de tabla

rename departamento to depto

En MySQL
rename table

Definir tablas: create table



Alter table: modificar tabla

create table

crea una tabla nueva y vacía
(sólo su definición)

alter table

modifica una tabla existente (su definición) que puede tener filas
(ejemplos: nueva columna, cambiar tipo a una columna, quitar una restricción not null...)

Alter table: add column

Añade una nueva columna a una tabla

```
alter table departamento  
add column oficina int;
```

column es
opcional

Alter table: drop column

Borra una columna existente

```
alter table departamento  
drop column nombre, oficina  
cascade;
```

column es
opcional

MySQL
drop nombre,
drop oficina

MySQL
no admite
cascade

Alter table: rename column

Cambia el nombre a una columna

```
alter table departamento  
rename column oficina to num_oficina;
```

column es
opcional

MySQL
change column
oficina num_oficina
integer ...

Ejercicios alter table

210. Añade el campo **ciudad** a la tabla **empleado**. Actualiza las filas de **empleado** para que **ciudad** contenga la ciudad de la dirección y dirección contenga sólo la calle y el número (suponer funciones `ciudad()` y `resto()` que obtienen a partir de una dirección, la ciudad, por un lado, y la calle y número por otro).

empleado NSS nombre apellidos fechaNcto direccion sexo salario jefe depto

not null *not null* **CE:empleado** **CE:departamento**

proyecto numero nombre localidad depto
unique CE:departamento

trabaja empleado proyecto horas

CE:empleado CE:proyecto

departamento numero nombre director fechaIniDir

unique CE:empleado
not null

Localidades depto localidad
CE: departamento

familiar empleado nombre fechaNcto sexo parentesco

CE: empleado

Alter table: modify

Cambia el tipo de datos a una columna

```
alter table departamento  
  modify column oficina char(5);
```

Puede dar problemas si
los valores existentes no
se acomodan al nuevo tipo

Alter table: alter...set default

Modifica el valor por defecto de una columna

```
alter table departamento  
  alter FechaInicJefe set default '1-1-1';
```


Alter table: alter...drop default

Borra el valor por defecto de una columna

```
alter table Departamento  
alter FechalnicJefe drop default;
```

Alter table: add constraint

Crea una nueva restricción en la tabla

```
alter table departamento  
  add constraint fk_jefe  
  foreign key(jefe)  
  references empleado(NSS)  
  on delete cascade  
  on update cascade;
```

Alter table: drop constraint

Borra una restricción de nombre conocido

```
alter table Departamento  
drop constraint fk_jefe;
```

MySQL
drop primary key...
drop foreign key...

Alter table: rename constraint

Renombra una restricción de nombre conocido

```
alter table departamento  
  rename constraint fk_jefe,  
  fk_superior;
```

Ejercicios DDL

211. Sobre la BD recién definida y sin filas, elimina la tabla **trabaja**.
212. Crea la tabla **jefe** y llénala con los empleados sin jefe. Deberá tener las mismas restricciones que **empleado**. Borra la columna jefe esta tabla.

empleado	<u>NSS</u>	nombre	apellidos	fechaNcto	direccion	sexo	salario	jefe	depto
		not null	not null						
								CE:empleado	CE:departamento

proyecto	<u>numero</u>	nombre	localidad	depto
		unique		CE:departamento

trabaja	<u>empleado</u>	<u>proyecto</u>	horas
		CE:empleado	CE:proyecto

departamento	<u>numero</u>	nombre	director	fechaIniDir
		unique	CE:empleado	not null

LocalidadesDepto	<u>depto</u>	<u>localidad</u>
		CE:departamento

familiar	<u>empleado</u>	<u>nombre</u>	<u>fechaNcto</u>	sexo	parentesco
					CE:empleado

Alter table: disable constraint

Desactiva una restricción de nombre conocido

```
alter table Departamento  
disable constraint pk_dpto cascade;
```

MySQL
para deshabilitar claves
set foreign_key_checks=0

Alter table: enable constraint

Reactiva una restricción de nombre conocido

```
alter table departamento  
enable constraint pk_dpto cascade;
```

MySQL
para habilitar claves foráneas
set foreign_key_checks=1

Redefinir tablas: alter table



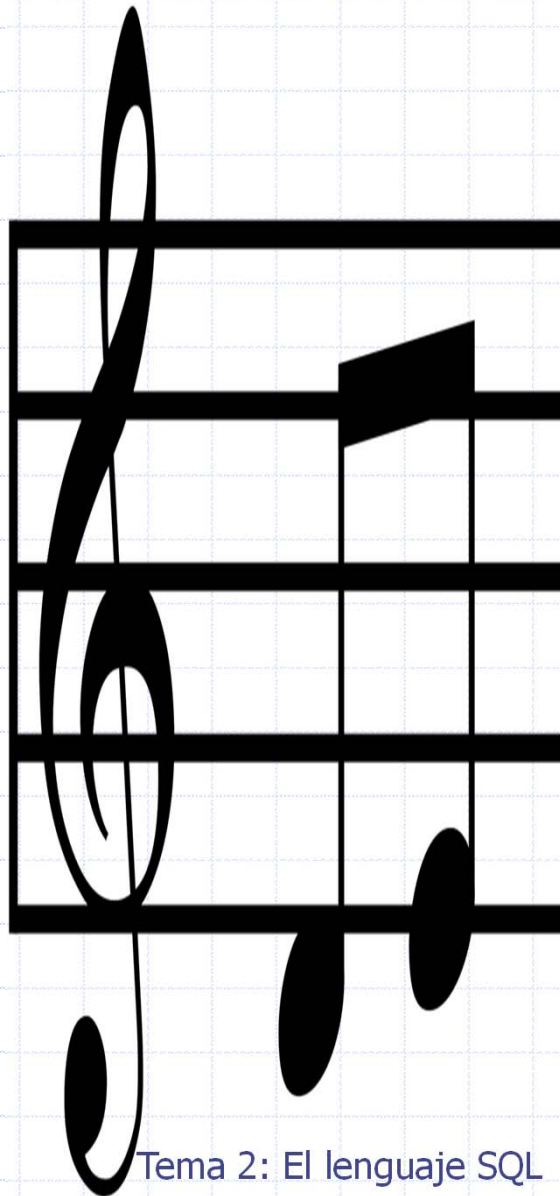
modify column

enable constraint
disable constraint

add column
drop column
rename column

alter col set default...
alter col drop default...

add constraint
drop constraint
rename constraint



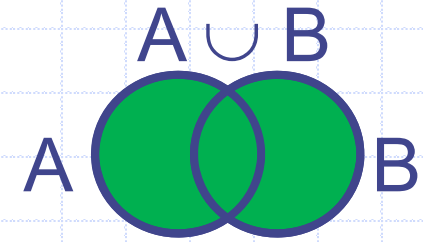
Índice

- ◆ Introducción
- ◆ LMD: consultas
- ◆ LDV
- ◆ LMD: actualizaciones
- ◆ LDD
- ◆ **LMD: más consultas**

Índice / Más consultas

- 4. **Union, intersect y except (all)**
- 5. Outer join, coalesce y case
- 6. Subconsultas de igual resultado
- 7. Subconsultas correlacionadas

Union



Sin filas
repetidas
(Díaz)

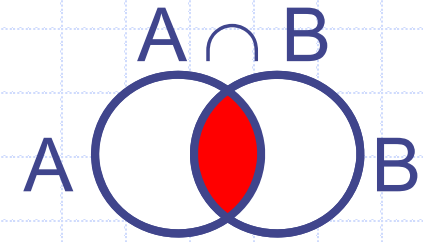
apell	loc
Díaz	Haro
Beltrán	Madrid
Lasa	Logroño
García	Madrid
Sánchez	Bilbao
Díaz	Logroño
García	Haro
Sánchez	Logroño

```
select apell  
from persona  
where loc='Logroño'  
union  
select apell  
from persona  
where loc='Haro'
```

apell
Díaz
García
Lasa
Sánchez

en
MySQL y
Access
sólo hay
union

Intersect



Sin filas
repetidas

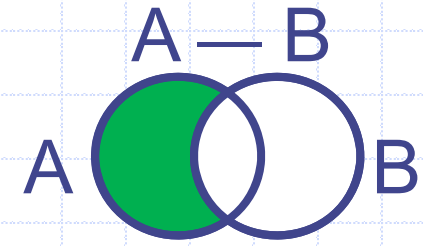
apell	loc
Díaz	Haro
Beltrán	Madrid
Lasa	Logroño
García	Madrid
Sánchez	Bilbao
Díaz	Logroño
García	Haro
Sánchez	Logroño

```
select apell
from persona
where loc='Logroño'
intersect
select apell
from persona
where loc='Haro'
```

apell

Díaz

Except



Sin filas
repetidas

apell	loc
Díaz	Haro
Beltrán	Madrid
Lasa	Logroño
García	Madrid
Sánchez	Bilbao
Díaz	Logroño
García	Haro
Sánchez	Logroño

```
select apell  
from persona  
where loc='Logroño'  
except  
select apell  
from persona  
where loc='Haro'
```

En
ORACLE
se llama
minus

apell
Lasa
Sánchez

Except no es conmutativa

Sin filas
repetidas

apell	loc
Díaz	Haro
Beltrán	Madrid
Lasa	Logroño
García	Madrid
Sánchez	Bilbao
Díaz	Logroño
García	Haro
Sánchez	Logroño

```
select apell  
from persona  
where loc='Haro'  
except  
select apell  
from persona  
where loc='Logroño'
```

apell
García

Nombres de columna diferentes

Se admiten

apell	loc
Díaz	Haro
Beltrán	Madrid
Lasa	Logroño
García	Madrid
Sánchez	Bilbao
Díaz	Logroño
García	Haro
Sánchez	Logroño

```
select loc
from persona
where loc like '%i%'
union
select apell
from persona
where loc like 'H%'
```

loc
Madrid
Bilbao
Díaz
García

Compatibilidad de unión: tipos

empleado

<u>cod</u>	nom	jefe
1	Miguel	2
2	Alicia	6
6	Lorena	null
7	Félix	8
8	Javier	6

depto

<u>num</u>	nom	dir
1	LSI	2
2	ATC	8
6	DMC	6

```
select cod
from empleado
where jefe is null
intersect
select nom
from depto
where dir is not null
```

da
ERROR

Compatibilidad de unión: numero

empleado

<u>cod</u>	nom	jefe
1	Miguel	2
2	Alicia	6
6	Lorena	null
7	Félix	8
8	Javier	6

depto

<u>num</u>	nom	dir
1	LSI	2
2	ATC	8
6	DMC	6

```
select cod, nom
from empleado
where jefe is null
intersect
select dir
from depto
where dir is not null
```

da
ERROR

ejercicios union/intersect/except

128. DNI de quienes son alumnos o profesores
129. Todos los datos de las personas que **no** son profesores
130. DNI de quienes **no** son profesores **ni** alumnos

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

not null CE:titulacion

unique *not null* *not null* *CE:profesor*
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Union all

empleado

<u>cod</u>	nom	apell
1	Félix	Pérez
2	Alicia	Balda
6	Lorena	Prat
7	Félix	Osés
8	Alicia	Balda

cliente

<u>cod</u>	nomb	apel
11	Iván	Tenés
12	María	López
6	Lorena	Prat
2	Alicia	Balda
15	Alicia	Balda

**cada repetición
cuenta como un
elemento diferente**

```
select nom, apell  
from empleado  
union all  
select nomb, apel  
from cliente  
where cod < 10
```

nom	apell
Félix	Pérez
Alicia	Balda
Lorena	Prat
Félix	Osés
Alicia	Balda
Lorena	Prat
Alicia	Balda

En
ORACLE
sólo hay
union all

Intersect all

empleado

<u>cod</u>	nom	apell
1	Félix	Pérez
2	Alicia	Balda
6	Lorena	Prat
7	Félix	Osés
8	Alicia	Balda

cliente

<u>cod</u>	nomb	apel
11	Iván	Tenés
12	María	López
6	Lorena	Prat
2	Alicia	Balda
15	Alicia	Balda

```
select nom, apell  
from empleado  
intersect all  
select nomb, apel  
from cliente
```

nom	apell
Alicia	Balda
Alicia	Balda
Lorena	Prat

cada repetición
cuenta como un
elemento diferente

Except all

empleado

<u>cod</u>	nom	apell
1	Félix	Pérez
2	Alicia	Balda
6	Lorena	Prat
7	Félix	Osés
8	Alicia	Balda

cliente

<u>cod</u>	nomb	apel
11	Iván	Tenés
12	María	López
6	Lorena	Prat
2	Alicia	Balda
18	Inés	Núñez

```
select nom, apell  
from empleado  
except all  
select nomb, apel  
from cliente
```

nom	apell
Félix	Pérez
Félix	Osés
Alicia	Balda

cada repetición
cuenta como un
elemento diferente

Ejercicios union/intersect/except

66. Identificadores de alumnos matriculados en 150212 y 130113 (ambas asign. a la vez)

Ejercicios 68, 81 y 82

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

not null CE:titulacion

unique *not null* *not null* *CE:profesor*
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Ejercicios union/intersect/except

131. Códigos de artículo del proveedor 'El Barato' sin ventas a clientes de 'Madrid'

cliente ident_client apell1 nombr_client nacion ciud

not null

tienda ident_tiend ciud nombr_gerent

not null

proveedor ident_prov nombr_prov

not null

articulo ident_artic nombr_artic peso color prec_compr prec_vent ident_prov

not null

not null not null **CE:proveedor**

venta ident_client ident_tiend ident_artic fech_vent cant_artics prec_total

CE:cliente CE:tienda CE:articulo

Índice / Más consultas

- 4. Union, intersect y except (all)
- 5. Outer join, coalesce y case**
- 6. Subconsultas de igual resultado
- 7. Subconsultas correlacionadas

Join y outer join

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

```
select *  
from   grupo as G  
       join  
       alum_grupo as AG  
       on G.num=AG.grupo
```

num	dia	hora	alum	grupo
1	lunes	12:00	111	1
1	lunes	12:00	333	1
2	jueves	16:00	222	2

Left [outer] join

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

```
select *  
from grupo as G  
→ left outer join  
alum_grupo as AG  
on G.num=AG.grupo
```

num	dia	hora	alum	grupo
1	lunes	12:00	111	1
1	lunes	12:00	333	1
2	jueves	16:00	222	2
3	jueves	18:00	null	null

Right [outer] join

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

```
select *  
from grupo as G  
→ right outer join  
alum_grupo as AG  
on G.num=AG.grupo
```

num	dia	hora	alum	grupo
1	lunes	12:00	111	1
1	lunes	12:00	333	1
2	jueves	16:00	222	2
null	null	null	444	4

Full [outer] join

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

select *

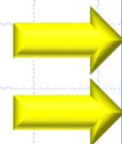
from

grupo as G

→ full outer join

alum_grupo as AG

on G.num=AG.grupo



num	dia	hora	alum	grupo
1	lunes	12:00	111	1
1	lunes	12:00	333	1
2	jueves	16:00	222	2
3	jueves	18:00	null	null
null	null	null	444	4

Ejercicios con outer join

132. Nombre y apellido de los alumnos sin asignaturas.
133. Id de asignaturas sin matriculaciones.

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null not null *CE:profesor* not null
 not null *CE:titulacion*

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejemplo 1 outer join - v1.0

Número, día, hora y número de alumnos de cada grupo (cero si no tiene alumnos)

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

```
select G.num, G.dia, G.hora, count(*) as alum
from grupo as G join alum_grupo as A
on G.num=A.grupo
group by G.num, G.dia, G.hora
```

num	dia	hora	alum
1	lunes	12:00	2
2	jueves	16:00	1

¡Falta el grupo 3!

Ejemplo 1 outer join - v1.1

Número, día, hora y número de alumnos de cada grupo (cero si no tiene alumnos)

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

```
select G.num, G.dia, G.hora, count(*) as alum
from grupo as G join alum_grupo as A
on G.num=A.grupo
group by G.num, G.dia, G.hora
```

union all

```
select G.num, G.dia, G.hora, 0
from grupo as G left join alum_grupo as A
on G.num=A.grupo
where A.alum is null
```

num	dia	hora	alum
1	lunes	12:00	2
2	jueves	16:00	1
3	jueves	18:00	0

Ejemplo 1 outer join – v2.1

Número, día, hora y número de alumnos de cada grupo (cero si no tiene alumnos)

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

```
select G.num, G.dia, G.hora, count(*) as alum
from grupo as G left join alum_grupo as A
on G.num=A.grupo
group by G.num, G.dia, G.hora
```

num	dia	hora	alum
1	lunes	12:00	2
2	jueves	16:00	1
3	jueves	18:00	1

¡Para el
grupo 3
cuenta
uno!

Ejemplo 1 outer join – v2.1

Número, día, hora y número de alumnos de cada grupo (cero si no tiene alumnos)

grupo

<u>num</u>	dia	hora
1	lunes	12:00
2	jueves	16:00
3	jueves	18:00

alum_grupo

<u>alum</u>	grupo
111	1
222	2
333	1
444	4

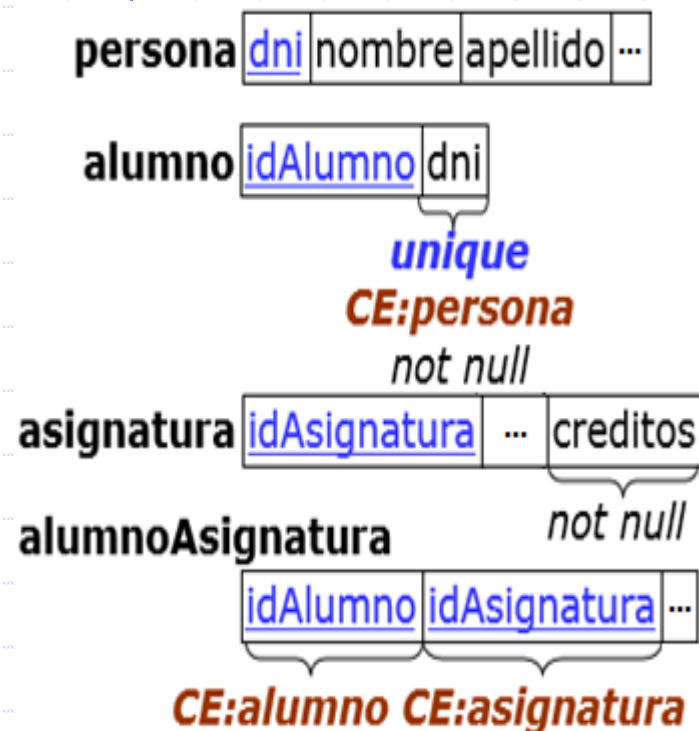
```
select G.num, G.dia, G.hora,  
       count(A.alum) as alum  
from grupo as G left join alum_grupo as A  
on G.num=A.grupo  
group by G.num, G.dia, G.hora
```

num	dia	hora	alum
1	lunes	12:00	2
2	jueves	16:00	1
3	jueves	18:00	0

**count(null)
da cero**

Ejemplo 2

Nombre y apellido de los alumnos junto al identificador de sus asignaturas. Deben salir también los alumnos sin asignaturas



```
select P.nombre, P.apellido, X.asignatura
from persona P join
alumno A on P.dni = A.dni left join
alumnoasignatura as X on A.Idalumno=
X.Idalumno
```

Ejemplo 3 outer join – v1.1

Nombre, apellido y nº de créditos en los que se ha matriculado cada alumno (cero cuando sea ninguno)

como el ejemplo 1
pero con más joins

persona dni nombre apellido ...

alumno idAlumno dni

unique

CE:persona
not null

asignatura idAsignatura ... creditos

alumnoAsignatura

not null

idAlumno idAsignatura ...

CE:alumno CE:asignatura

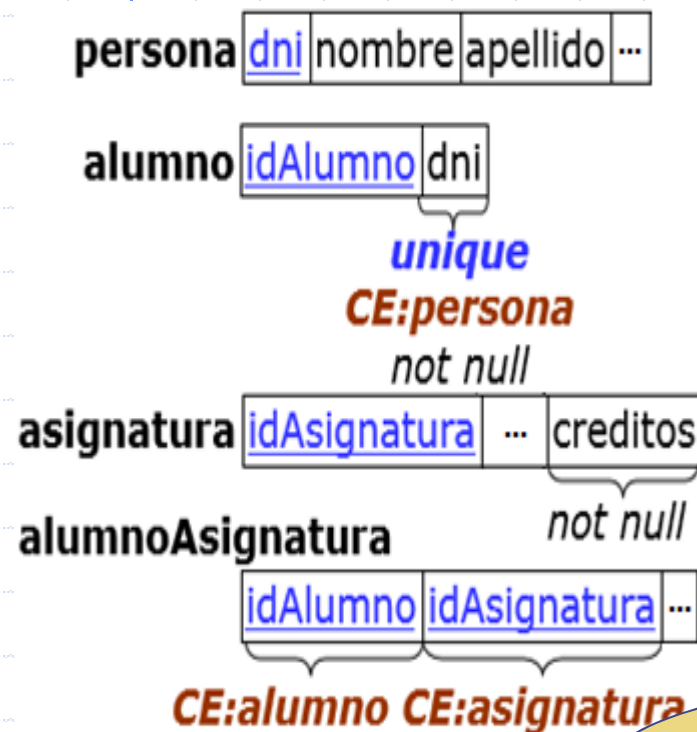
```
select P.nombre, P.apellido, sum(A.creditos)
from persona P join
alumno A on P.dni = A.dni left join
alumnoasignatura as X on A.Idalumno=
X.Idalumno join
asignatura as G on X.IdAsignatura =
G.IdAsignatura
group by A.IdAlumno, P.nombre, P.apellido
```

¡ está mal !

¡ las filas alumnoasignatura
con nulos no emparejan
con asignatura (join) !

Ejemplo 3 outer join – v1.2

Nombre, apellido y nº de créditos en los que se ha matriculado cada alumno (cero cuando sea ninguno)



```
select P.nombre, P.apellido, sum(A.creditos)
from (persona P join
alumno A on P.dni=A.dni) left join
(alumnoasignatura as X join
asignatura as G on X.IdAsignatura =
G.IdAsignatura) on A.Idalumno =
X.Idalumno
group by A.IdAlumno, P.nombre, P.apellido
```

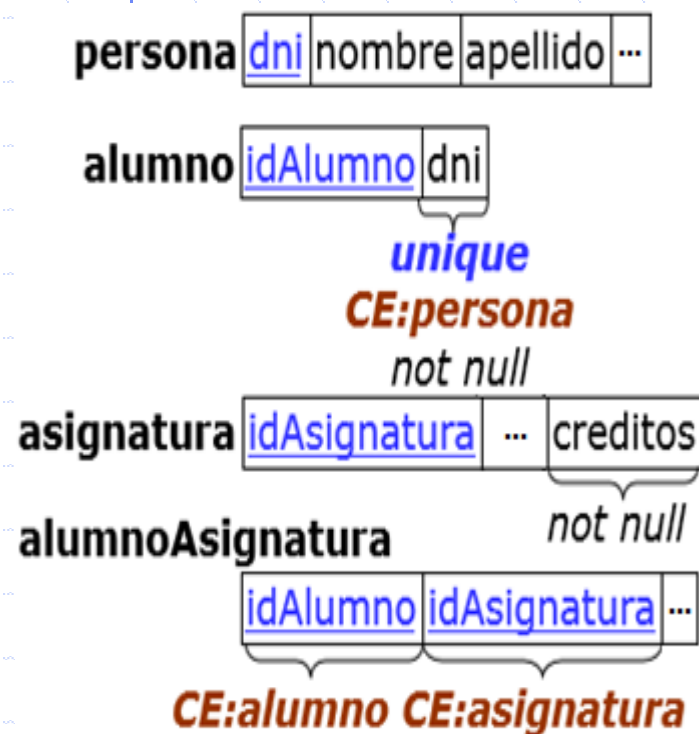
¡Los paréntesis
de los join son
importantes!

¡ no sale bien porque
sum(null) da null !

Ejemplo 3 outer join – v1.3

coalesce

Nombre, apellidos y nº de créditos en los que se ha matriculado cada alumno (cero cuando sea ninguno)



select P.nombre, P.apellido,
coalesce(sum(A.créditos), 0)
from (persona P join
alumno A on P.dni=A.dni) **left join**
(alumnoasignatura as X join
asignatura as G on X.IdAsignatura =
G.IdAsignatura) **on** A.Idalumno =
X.Idalumno
group by A.IdAlumno, P.nombre, P.apellido

corregimos
sum(**null**)

Ejercicios outer join/coalesce

134. Nombre, apellido y nº de créditos matriculados de los alumnos, incluyendo los de 0 créditos.
135. Nombre de titulaciones, nº de asignaturas que tienen y nº total de créditos, incluyendo las titulaciones sin asignaturas con sendos ceros.

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

unique *not null* *not null* *CE:profesor*
not null *CE:titulacion*

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno *CE:asignatura* *Nº de veces que se ha matriculado*

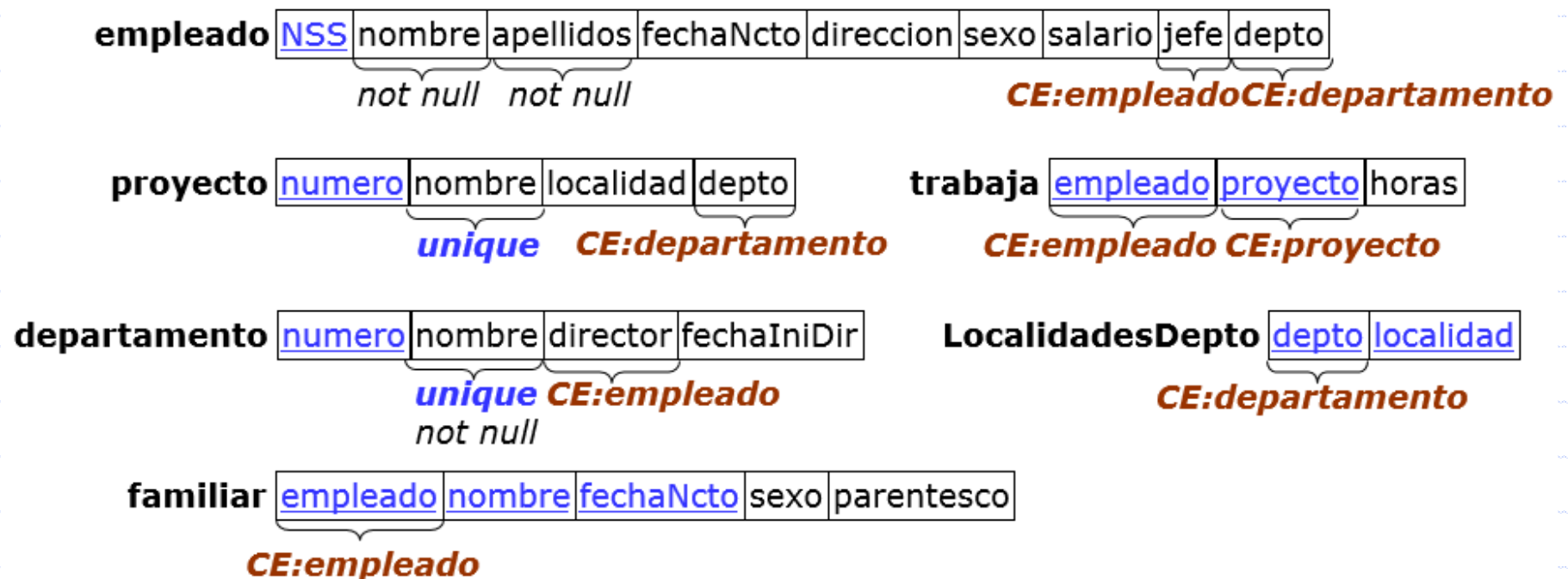
Ejercicios con coalesce y case

136. Para cada asignatura obtener su código, nombre de asignatura y nombre de titulación. Si no tiene titulación poner 'Asignatura de Campus' en lugar del código de titulación



Ejercicios con coalesce y case

137. NSS, salario, salario total de los subordinados (salarioSub) y nº de subordinados (numSub) de todos los empleados



- 

CE:lector

CE:autor CE:libro

not null

CE:lector
not null

Outer join: ¿on o where?

Nombres de **todas** las asignaturas del turno de tarde junto a sus alumnos (las que los tengan)

asig

<u>id</u>	nom	hora
1	BD	12:00
2	TP	16:00
3	EC	18:00
4	null	17:30

```
select A.nom, AA.alum
from asig A left join
aluAsig AA on
A.id=AA.asig
where A.hora>=15:00
```

aluAsig

<u>alum</u>	<u>asig</u>
111	1
222	1
333	3

1º left join y
luego where
(BD lo borra
where)

nom	alum
TP	null
EC	333
null	null

null rojos
creados
por left join

Outer join: ¿on o where?

Nombres de **todas** las asignaturas del turno de tarde junto a sus alumnos (las que los tengan)

asig

<u>id</u>	nom	hora
1	BD	12:00
2	TP	16:00
3	EC	18:00
4	null	17:30

aluAsig

<u>alum</u>	<u>asig</u>
111	1
222	1
333	3

```
select A.nom, AA.alum
from asig A left join
aluAsig AA on
A.id=AA.asig and
A.hora>=15:00
```

Se ha
pasado la
condición
al on

nom	alum
BD	null
TP	null
EC	333
null	null

Ahora salen
TODAS las
filas de asig
¿por qué?

¿ Qué es correcto *on* o *where*?

Nombre y nº de **asignaturas de 2º** de todas las **titulaciones con id entre 30-55** (algunas sin 2º implan.)

```
select t.nombre, count(a.IdAsignatura)
from titulacion t left join
    asignatura a on t.idtitulacion=a.idtitulacion and ...
where ...
group by t.nombre
```

recuerda:
1º hace **left join**
y luego **where**

- 1) **a.curso=2** en **where**, **t.IdTitulacion between 30 and 55** en **where**
- 2) **a.curso=2** en **on**, **t.IdTitulacion between 30 and 55** en **on**
- 3) **a.curso=2** en **on**, **t.IdTitulacion between 30 and 55** en **where**
- 4) **a.curso=2** en **where**, **t.IdTitulacion between 30 and 55** en **on**

Ejercicios con outer join

141. Nombre de editorial, nº de libros de más de 1.000 páginas y nº total de páginas de los libros de editoriales de Barcelona.

libro id titulo editorial año peso paginas precio primer_solicitante

CE:editorial

editorial nombre direccion tfno

CE:lector

autor id nombre apellidos

autorlibro autor libro

CE:autor CE:libro

lector id nombre apellidos direccion tfno fecha_inscripcion

biblioteca id nombre direccion tfno

ejemplar libro ejemplar biblioteca fecha_solicitud fecha_compra

CE:libro

CE:biblioteca

not null

prestamo libro ejemplar fecha_prestado fecha_devuelto lector

CE:ejemplar

CE:lector

not null

Ejercicios con outer join

142. ID y nombre de todas las bibliotecas de Logroño, n° de ejemplares con peso mayor de 980 gr. (NumLibros) y peso total de esos ejemplares (PesoLibros).



Ejercicios con outer join

143. Nombre y dirección de editoriales de Barcelona y Valencia, n° de libros editados de más de 900 páginas y n° total de páginas de esos libros. Ordena el resultado descendientemente por n° de libros de más de 900 páginas.

libro	<u>id</u>	titulo	editorial	año	peso	paginas	precio	primer_solicitante
			CE:editorial					CE:lector

editorial	<u>nombre</u>	direccion	tfno
-----------	---------------	-----------	------

autor	<u>id</u>	nombre	apellidos
-------	-----------	--------	-----------

autorlibro	<u>autor</u>	<u>libro</u>
	<i>CE:autor CE:libro</i>	

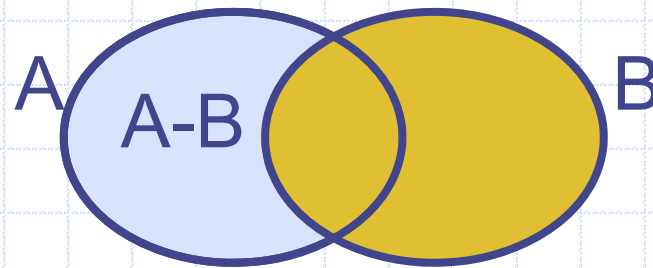
lector	<u>id</u>	nombre	apellidos	direccion	tfno	fecha_inscripcion
--------	-----------	--------	-----------	-----------	------	-------------------

biblioteca	<u>id</u>	nombre	direccion	tfno
------------	-----------	--------	-----------	------

ejemplar	<u>libro</u>	<u>ejemplar</u>	biblioteca	fecha_solicitud	fecha_compra
	<i>CE:libro</i>		<i>CE:biblioteca</i> not null		

prestamo	<u>libro</u>	<u>ejemplar</u>	<u>fecha_prestado</u>	fecha_devuelto	lector
	CE:ejemplar				CE:lector not null

Calculando la diferencia (A-B)



- ◆ A veces se puede con $< >$ en where
- ◆ Con except
- ◆ Con outer join
- ◆ Con subconsultas y not in / not exists

Usando <> en when

Identificador de las asignaturas **sin** alumnos

asig

<u>id</u>	nom	hora
1	BD	12:00
2	TP	16:00
3	EC	18:00

aluAsig

<u>alum</u>	<u>asig</u>
111	1
222	1
333	3

```
select A.id
from asig as A join
aluAsig as AA on
A.id <> AA.idAsig
```

¿Qué
ocurre?
(veamos
el join
completo)

id
1
2
2
2
3
3

Usando <> en when: join completo

Identificadores de las asignaturas **sin** alumnos

asig

<u>id</u>	nom	hora
1	BD	12:00
2	TP	16:00
3	EC	18:00

```
select *  
from asig as A join  
aluAsig as AA on  
A.id <> AA.asig
```

aluAsig

<u>alum</u>	<u>asig</u>
111	1
222	1
333	3

id	nom	hora	alum	asig
1	BD	12:00	333	3
2	TP	16:00	111	1
2	TP	16:00	222	1
2	TP	16:00	333	3
3	EC	18:00	111	1
3	EC	18:00	222	1

Así no podemos
calcular la
diferencia

Usando except

Identificadores de las asignaturas sin alumnos

asig

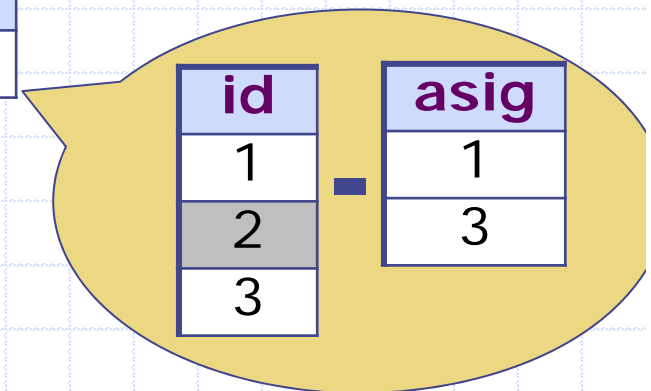
<u>id</u>	nom	hora
1	BD	12:00
2	TP	16:00
3	EC	18:00

aluAsig

<u>alum</u>	<u>asig</u>
111	1
222	1
333	3

```
select id
from asig
except
select asig
from AluAsig
```

id
2



Ejemplo donde funciona <>

Identificadores de asignaturas
sin cañón de vídeo operativo

asig

<u>id</u>	nom	hora	video
1	BD	12:00	sí
2	TP	16:00	no
3	EC	18:00	sí
4	F	17:00	reparando
5	L	9:00	solicitado

```
select id
from asig
where video <> 'sí'
```

id
2
4
5

Otro usando except

Nombres de los cuidadores que **no** sean nombres de profesores

cuid

<u>asig</u>	<u>N</u>	cuid
1	1	Pepe
1	2	Ana
2	1	Pepe
2	2	Luisa
3	1	null

profe

<u>id</u>	nom
1	Pepe
2	Ana
3	Rosa
4	Juan

N es un nº
correlativo de
cuidador en
una asignatura

```
select cuid  
from cuid  
except  
select nom  
from profe
```

cuid
Luisa
null

¿Cómo
eliminamos
el null?

Otro usando except / sin nulos

Nombres de los cuidadores que **no** sean nombres de profesores

cuid

<u>asig</u>	<u>N</u>	cuid
1	1	Pepe
1	2	Ana
2	1	Pepe
2	3	Luisa
3	1	null

profe

<u>id</u>	cuid
1	Pepe
2	Ana
3	Rosa
4	Juan

```
select cuid
from cuid
where cuid is not null
except
select cuid
from profe
```

cuid
Luisa

Ahora con outer join

Nombres de los cuidadores que **no** sean nombres de profes

cuid

<u>asig</u>	<u>N</u>	cuid
1	1	Pepe
1	2	Ana
2	1	Pepe
2	3	Luisa
3	1	null

profe

<u>id</u>	cuid
1	Pepe
2	Ana
3	Rosa
4	Juan

```
select C.cuid
from cuid C left join
      profe P on
              C.cuid=P.cuid
where  P.id is null
```

cuid
Luisa
null

Otra
vez el
null

Ahora con outer join / sin nulos

Nombres de los cuidadores que **no** sean nombres de profes

cuid

<u>asig</u>	<u>N</u>	cuid
1	1	Pepe
1	2	Ana
2	1	Pepe
2	3	Luisa
3	1	null

profe

<u>id</u>	cuid
1	Pepe
2	Ana
3	Rosa
4	Juan

```
select C.cuid
from cuid C left join
      profe P on
           C.cuid=P.cuid
where  P.id is null and
       C.cuid is not null
```

cuid
Luisa

¿Funcionan igual en este caso?

Nombres de los cuidadores que **no** sean nombres de profes

cuid

<u>asig</u>	<u>N</u>	cuid
1	1	Pepe
1	2	Ana
2	1	Pepe
2	3	Luisa
3	1	null

profe

<u>id</u>	cuid
1	Pepe
2	Ana
3	Rosa
4	Juan
5	null



```
select cuid
from cuid
except
select cuid
from profe
```

```
select C.cuid
from cuid C left join
profe P on
C.cuid=P.cuid
where P.id is null
```

cuid

Luisa

aquí un
nulo
elimina al
otro

cuid

Luisa

null

Conjuntos y outer join

**union
intersect
except**



**union all
intersect all
except all**



**left join
right join
full join**

**coalesce
case**

Índice / Más consultas

- 4. Union, intersect y except (all)
- 5. Outer join, coalesce y case
- 6. Subconsultas de igual resultado**
- 7. Subconsultas correlacionadas

Una subconsulta

Asignaturas con menos créditos que la media

```
select *  
from asignatura  
where credits <
```

¡va entre
paréntesis!

```
( select avg(credits)  
  from asignatura )
```

es una select
dentro de
otra select

media de
créditos por
asignatura

Una subconsulta

Asignaturas con menos créditos que la media

```
select *  
from asignatura  
where creditos <
```

deben conectarse
bien consulta y
subconsulta
(comparar número
con número)

Para cada
fila de
asignatura la
subconsulta
¡devuelve el
mismo
valor!

```
( select avg(creditos)  
  from asignatura )
```

puede haber
más niveles:
aquí dentro
otra select...

Dónde colocar subconsultas

- ◆ En el estándar sólo en **where** y **having**
- ◆ Los SGBD también las admiten en:
 - **from**
 - **select**
 - **when** de **case**
 - **coalesce**
 - dentro de funciones agregadas
 - ...

Subconsulta con <, >, <=, >=, =, <>

Asignaturas con menos créditos que la media

```
select *  
from asignatura  
where credits <
```

para poder
comparar con
créditos debe dar
**¡ siempre 1 fila y
1 columna ¡**

```
( select avg(credits)  
  from asignatura )
```

Aquí la
subconsulta
está en la
condición
where

Subconsultas de varias columnas

Clientes que superen **en una venta** a la media de precio total y la de cantidad de artículos

```
select distinct ident_client  
from venta  
where (prec_total, cant_artics) >
```

compara 2 pares
(número, número)

Access y
SqlServer
no lo
permiten

```
(select avg(prec_total),  
         avg(cant_artics)  
from venta  
);
```

¡ devuelve 2
columnas !

Ejercicios con <, >, <=, >=, =, <>

51: Listado de asignaturas con más créditos que Seguridad Vial (asig. de libre elección)

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion credits cuatrimestre costeBasico idProfesor curso
unique not null *CE:titulacion* not null not null *CE:profesor* not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios con <, >, <=, >=, =, <>

67: Nombres de asignatura de la titulación 130110 con coste básico mayor que el coste básico medio de la titulación

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

not null CE:titulacion
unique
not null *not null*
CE:profesor
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Subconsultas con [not] in (\in / \notin)

Nombres de las asignaturas sin alumnos

```
select nombre  
from asignatura  
where idAsignatura not in  
( select idAsignatura  
  from alumnoAsignatura  
  )
```

not in
es \notin

Identificadores
de asignaturas
con alumnos

¡devuelve
varias filas y
una columna!

Proceso de subconsulta con in (€)

Nombre y cuota básica de clubes cuya cuota básica sea distinta a cualquiera de las pagadas por Luis

socio

<u>nom</u>	<u>club</u>	cuota
Ana	c3	20
Juan	c3	25
Luis	c1	20
Luis	c2	20
Luis	c3	15

club

<u>nom</u>	cBasica
c1	10
c2	15
c3	20
c4	10

cuota
20
20
15

```
select C.nom, C.cBasica
from club C
where C.cBasica not in
(select S.cuota
 from socio S
 where S.nom='Luis')
```

nom	cBasica
c1	10
c4	10

Cuotas de Luis

Nulos con except y not in /1

R

<u>id</u>	R1
1	a1
2	a2
3	a2
4	a3
5	null

S

<u>id</u>	S1
1	a1
2	a2
3	a4
4	a5

```
select R1
from R
except
select S1
from S
```

R1
a3
null

```
select R1
from R
where R1 not in
(select S1
from S
)
```

R1
a3

¿por qué no
está el null?

Nulos con except y not in /2

R

<u>id</u>	R1
1	a1
2	a2
3	a2
4	a3
5	null

S

<u>id</u>	S1
1	a1
2	a2
3	a4
4	a5
5	null

```
select R1
from R
except
select S1
from S
```

R1
a3

```
select R1
from R
where R1 not in
(select S1
from S
)
```

R1

¿por qué vacío?

Ejercicios con in

- 66 Id de alumnos con las asignaturas 150212 y 130113 (hecho con join y having)
- 69 Nombres de titulación con alguna asignatura de más de 4,5 créditos (hecho con join)

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion credits cuatrimestre costeBasico idProfesor curso
unique not null not null *CE:profesor* not null
 not null *CE:titulacion*

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios con in

18. Nombre y apellido de los alumnos que tengan cualquier asignatura menos la 150 ó la 130.

¿La siguiente consulta resuelve el problema?

```
select p.nombre, p.apellido
from persona as p join
      alumno as a on p.dni=a.dni join
      alumnoAsignatura as aa on
           a.idalumno=aa.idalumno
where aa.idAsignatura not in ('150', '130')
```


Subconsultas con **all** (\forall) y **any** (\exists)

Nombre/s de la asignatura/s con más alumnos

```
select A.nombre  
from asignatura A join  
      alumnoAsignatura B on A.idasignatura=B.idasignatura  
group by A.idasignatura, A.nombre  
having count(*) >=all
```

Para cada grupo
compara un número con
un conjunto de números

Número de
alumnos en
cada asignatura

```
( select count(*)  
  from alumnoAsignatura  
  group by A.idasignatura  
)
```

varias filas
1 columna

Comparaciones all (\forall) y any (\exists)

having count(*) \geq all (select count(*) from ...)

Comparan el valor (**único**) de su izquierda con los devueltos por la subconsulta de la derecha que deben ser del **mismo tipo**

\geq all, \leq all, =all, \neq all, ...
 \geq any, \leq any, =any, \neq any, ...

Comparaciones all (\forall) y any (\exists)

having count(*) **>=all** (select count(*) from

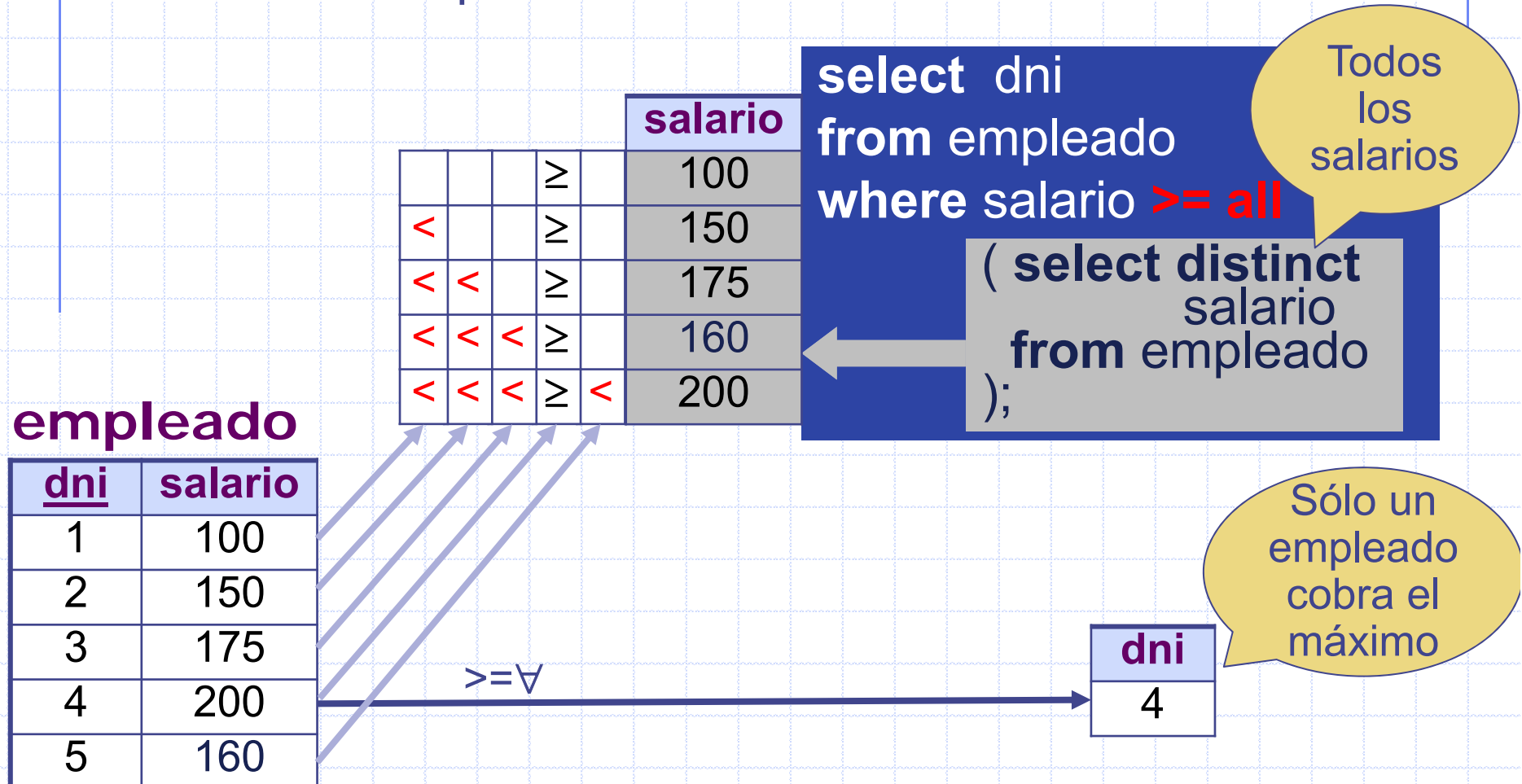
True si la condición ($>=$) es cierta para **todos** los valores devueltos por select

having count(*) **>=any** (select count(*) from

True si la condición ($>=$) es cierta para **alguno** de los valores devueltos

Proceso de subconsulta con all (∀)

DNI de los empleados con el máximo de los sueldos



Equivalencia con all (\forall)

where A **<>all** (select B from ...)



where A **not in** (select B from ...)

Equivalencia con any (\exists)

where A **=any** (select B from ...)



where A **in** (select B from ...)

¿Son ciertas?

where not (A **in** (select B from ...))



where A **not in** (select B from ...)

where not (A **in** (select B from ...))



where not(A **=any** (select B from ...))

where A **not in** (select B from ...)



where A **<>any** (select B from ...)

Subconsultas con [not] exists ($\neq \emptyset$)

Id de las asignaturas de la titulación 117 si aun no hay alumnos matriculados en esa titulación

```
select A.idAsignatura  
from asignatura A  
where A.idTitulación='117' and  
not exists
```

$= \emptyset$

¡ A la izquierda de **not exists** no se pone nada !

Id de alumnos en la titulación 117

```
( select idAlumno  
from alumnoAsignatura as AA join  
asignatura as A on  
AA.idAsignatura=A.idAsignatura  
where A.idTitulación=117
```


Subconsulta vacía con exists, any y all

exists (\emptyset) = FALSE

not exists (\emptyset) = TRUE

A >=any (\emptyset) = FALSE

A >=all (\emptyset) = TRUE !

Ejercicios all, any y exists

144. Nombres de asignatura con mayor nº de créditos (con y sin max en subconsulta).

¿Sirve esto?:

```
select nombre, max(créditos)
from asignatura group by nombre
```

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null not null *CE:profesor* not null
 not null *CE:titulacion*

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios all, any y exists

145. Nombres de asignatura exceptuando las de mayor nº de créditos

74. Personas domiciliadas en una de las localidades con más personas de la BD

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

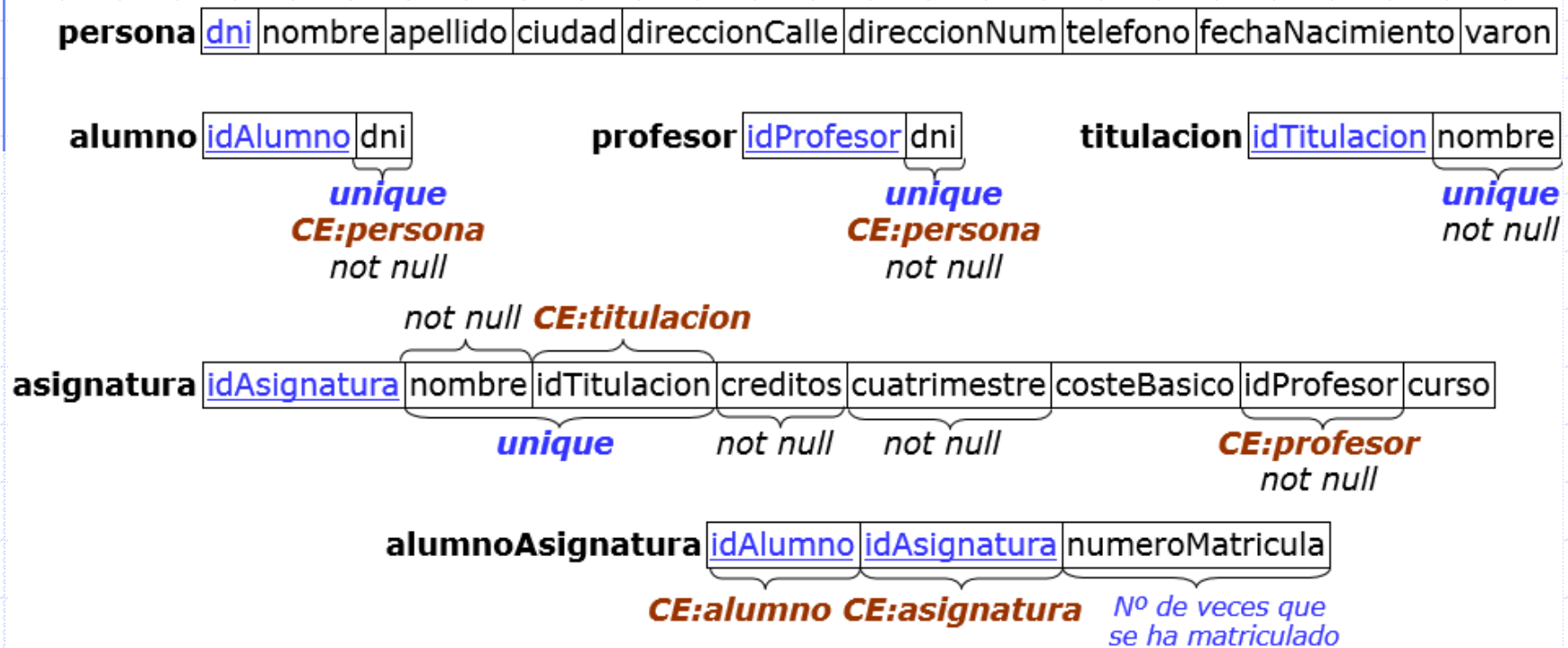
titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null *CE:titulacion* not null not null *CE:profesor* not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura Nº de veces que se ha matriculado

Ejercicios all, any y exists

146. Personas de la BD no domiciliadas en alguna de las localidades con más personas



Subconsultas

valor > (subconsulta)
<, =, <>, >=...

valor in (subconsulta)
valor not in (subcons.)

exists (subconsulta)
not exists (subconsulta)

valor >=all (subconsulta)
=all, <>all, <all...

valor >=any (subconsulta)
=any, <>any, <any...



Índice / Más consultas

- 4. Union, intersect y except (all)
- 5. Outer join, coalesce y case
- 6. Subconsultas de igual resultado
- 7. Subconsultas correlacionadas**

Subconsulta correlacionada

DNI de los empleados que más ganan de **su** departamento

```
select E.dni  
from empleado E  
where E.salario >= ALL
```

```
( select ED.salario  
  from empleado ED  
  where ED.depto=E.depto -- correlación  
);
```

Salarios de los
empleados de **SU**
departamento

Subconsulta correlacionada

DNI de los empleados que más ganan de su departamento

```
select E.dni  
from empleado E  
where E.salario >= ALL
```

```
( select ED.salario  
  from empleado ED  
  where ED.depto=E.depto -- correlación  
);
```

Correlación:
referencia a
un campo de
la consulta
principal

La misma con comparación =

DNI de los empleados que más ganan de su departamento

```
select E.dni  
from empleado E  
where E.salario =
```

Salario
máximo del
depto de E

```
( select max(ED.salario)  
  from empleado ED  
  where ED.depto=E.depto -- correlación  
);
```

La misma con not exists ($=\emptyset$)

DNI de los empleados que más ganan de su departamento

```
select E.dni  
from empleado E  
where not exists
```

```
( select *  
  from empleado ED  
  where ED.depto=E.depto -- correlación  
        and ED.salario>E.salario -- correlación  
);
```

Empleados
del depto de
E que ganan
más que él

Proceso de exists con correlación

nota Nombre de los alumnos sin suspensos

<u>dni</u>	asig	nota
1	7984	4
1	7450	4,5
1	7540	8,5
2	7984	6
2	4544	3
3	7984	7,5
3	4544	9
3	7540	8

alumno

<u>dni</u>	nombre
1	Jon
2	Ana
3	Begoña

```
select A.nombre  
from alumno A  
where not exists
```

```
(select N.asig  
from nota N  
where N.nota<5 and  
N.dni=A.dni)
```

asig	asig	asig
7984	4544	
7450		

nombre
Begoña

Suspensos
de A

=∅

Ejercicios correlaciones

147. Nombre de las asignaturas de **Matemáticas** con coste básico mayor al coste b. medio por asignatura de **la** titulación (sin correlación)
148. Nombre de las asignaturas con coste básico mayor al coste básico medio por asignatura de **su** titulación



Ejercicios correlaciones

149. DNI de los alumnos de Madrid con más créditos de entre todos los alumnos de Madrid (sin correlación)
150. DNI y localidad de los alumnos con más créditos de entre todos los alumnos de **su** misma localidad



Un in (∈) con correlación

Nombres de asignatura con coste básico
único en su titulación

```
select A.nombre  
from asignatura A  
where A.costeBasico not in
```

```
( select A1.costeBasico  
  from asignatura A1  
  where A1.idTitulación = A.idTitulación --corr.  
    and A1.idAsignatura <> A.idAsignatura --corr.  
);
```

costes básicos
de asignaturas
de la titulación de
A salvo el de A

Proceso de in (€) con correlación

Nombre de los clubes donde ningún socio paga la cuota básica del club

socio

<u>nom</u>	<u>club</u>	cuota
Ana	c3	20
Juan	c3	25
Luis	c3	15
Luis	c1	20
Luis	c2	20

club

<u>nom</u>	cBasica
c4	10
c3	20
c2	15
c1	10

```
select C.nom, C.cBasica
from club C
where C.cBasica not in
```

```
(select S.cuota as cu
from socio S
where S.club=C.nom)
```

cu	cu	cu	cu
20	20	20	20
25			
15			

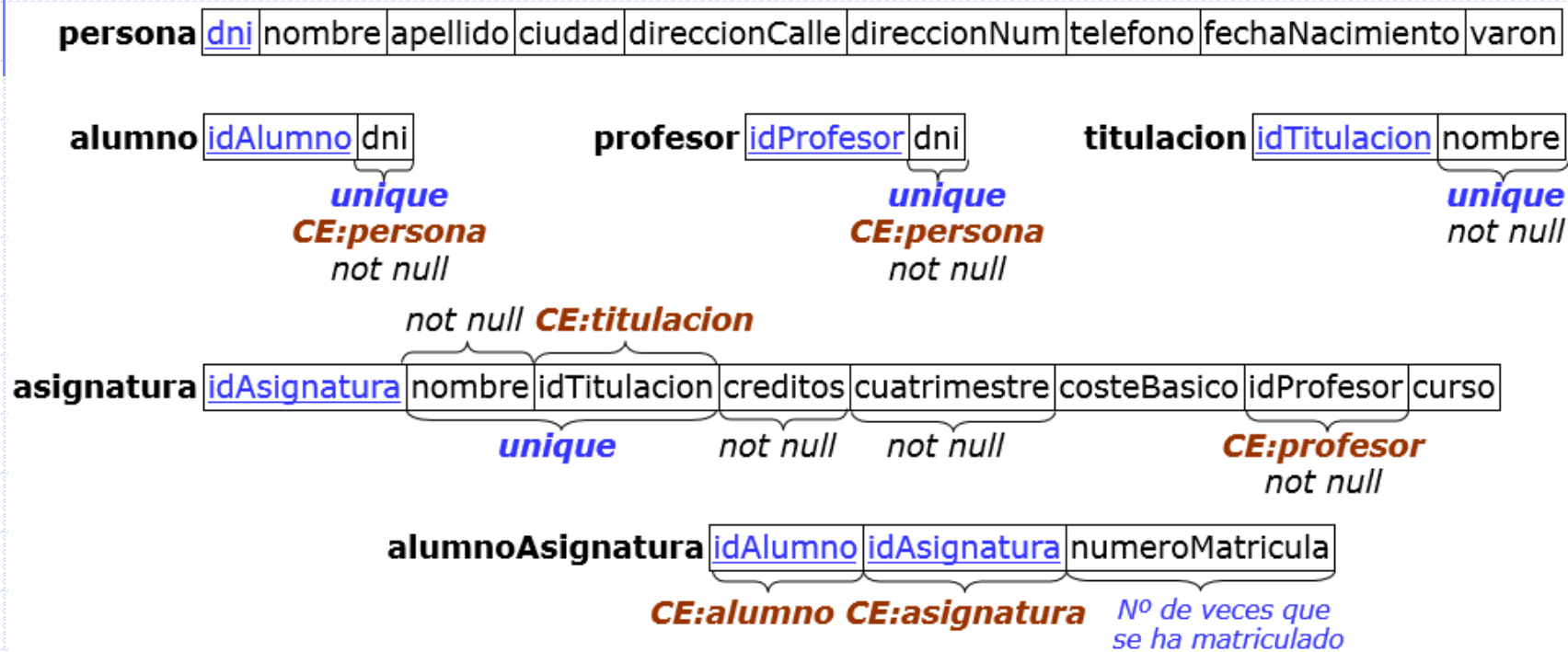
nom	cBasica
c4	10
c2	15
c1	10

Cuotas de C

Ejercicios correlaciones

68. Listado de asignaturas sin alumnos (con in y con exists)

151. Listado de las asignaturas que tienen a todos los alumnos de la BD matriculados ; **DIVISIÓN !**



División relacional (÷)

NSS de quienes trabajan **en todos** los proyectos de Logroño

trabaja

<u>NSS</u>	proy
111	1
111	2
333	1
333	2
333	3
444	1
444	4
555	1

proyecto

<u>proy</u>	loc
1	Logroño
2	Logroño
3	Logroño
4	Soria

trabaja ÷ proyecto

<u>NSS</u>
333

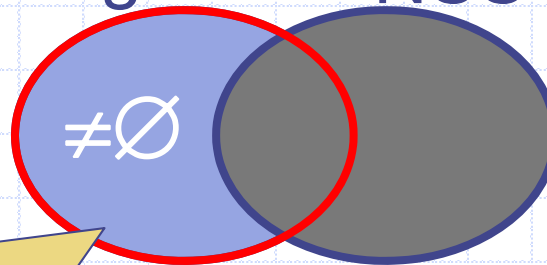
¿Cómo podemos hacerlo en SQL?

todos los proyectos de Logroño

División en SQL: visión de conjuntos

NSS de quienes trabajan **en todos**
los proyectos de Logroño

Proyectos de Logroño Proyectos de NSS=333



resta no vacía \Rightarrow **NO**
tiene todos los de
Logroño

Buscamos los que
 \bigcirc except $\bullet = \emptyset$

¿Cómo lo ponemos en SQL?

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select E.NSS  
from empleado as E  
where
```

Buscamos los que

 except  = \emptyset

not exists (

```
select  
from  
where
```

except

```
select  
from  
where
```

)

¿Cómo lo ponemos en SQL?

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select E.NSS  
from empleado as E  
where
```

not exists (

```
select P.número  
from proyecto as P  
where P.localidad='Logroño'
```

except

```
select  
from  
where
```

)

proyectos
de
Logroño

¿Cómo lo ponemos en SQL?

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select E.NSS  
from empleado as E  
where
```

not exists (

```
select P.número  
from proyecto as P  
where P.localidad='Logroño'
```

except

```
select T.proyecto  
from trabaja as T  
where T.NSS=E.NSS --correlación
```

proyectos
en los
que
trabaja E

)

Visión de conjuntos con not in

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select E.NSS  
from empleado as E  
where
```

```
not exists (
```

```
select *  
from proyecto as P  
where P.localidad='Logroño' and  
P.numero not in
```

```
(select T.proyecto  
from trabaja as T  
where T.NSS=E.NSS) --correlación
```

```
)
```

Visión del cálculo relacional

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select E.NSS  
from empleado as E  
where
```

```
not exists (
```

```
select *  
from proyecto as P  
where P.localidad='Logroño' and  
not exists
```

```
(select *  
from trabaja as T  
where T.NSS=E.NSS and --correlación  
T.proyecto=P.numero) --correlación
```

Cambia el **not in** anterior por un 2º **not exists** con una nueva correlación

```
)
```

Visión de conjuntos contando

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select T.NSS  
from trabaja as T  
group by T.NSS  
having count(*) =
```

Número de
proyectos de
T.NSS

Número de
proyectos de
Logroño

```
( select count(*)  
from proyecto as P  
where P.localidad='Logroño' )
```

¿Por qué
está mal?

Cuidado con lo que se cuenta

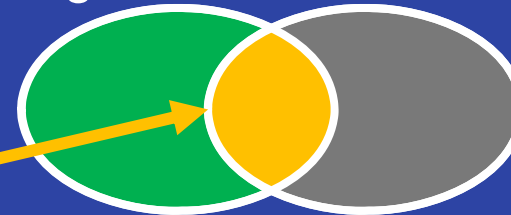
NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select T.NSS  
from trabaja as T  
group by T.NSS  
having count(*) =
```

```
( select count(*)  
  from proyecto as P  
  where P.localidad='Logroño' )
```

Proyectos de
Logroño

Proyectos de
NSS=333



Debería contar sólo el
área naranja y también
cuenta el área gris

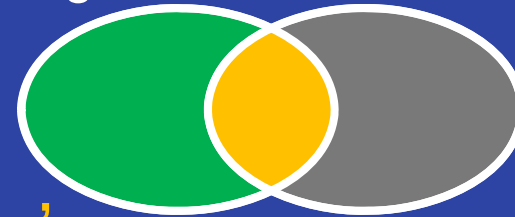
Contando bien

NSS de quienes trabajan **en todos**
los proyectos de Logroño

```
select T.NSS
from trabaja as T join
    proyecto as P on
        T.np=P.numero
where P.localidad='Logroño'
group by T.NSS
having count(*) =
```

Proyectos de
Logroño

Proyectos de
NSS=333



```
( select count(*)
  from proyecto as P
  where P.localidad='Logroño' )
```

Conclusión: saberse un esquema

AAA de A de quienes CCC en todos los BBB

atributos
AAA de A

```
select  
from A  
where
```

Buscamos los que
● except ● = \emptyset

todos los
BBB

not exists (

```
select  
from  
where
```

except

CCC
correspondiente
al A actual

```
select  
from  
where
```

--correlación

Resolviendo divisiones

152. ID de las asignaturas con todos los alumnos que a su vez son profesores

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni

unique
CE:persona
not null

profesor idProfesor dni

unique
CE:persona
not null

titulacion idTitulacion nombre

unique
not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso

not null CE:titulacion

unique

not null

not null

CE:profesor
not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula

CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Resolviendo divisiones

- ◆ Identificadores de las asignaturas en las que están matriculados todos los alumnos que a su vez son profesores.

División!!

Identificadores de las asignaturas para las que no exista un alumno que a su vez sea profesor que no esté matriculado.

Resolviendo divisiones

Identificadores de las asignaturas para las que no exista un alumno que a su vez sea profesor que no esté matriculado.

```
select idAsignatura  
from asignatura A  
where not exists
```

```
(select *  
from alumno Al natural join profesor P  
where Al.IdAlumno not in
```

Alumnos que son profesores

```
(select AA.IdAlumno  
from alumnosasignatura  
where A.idasignatura = AA.idasignatura -- correlación)
```

Alumnos matriculados en la
asignatura (correlacionada)

Resolviendo divisiones

Identificadores de las asignaturas para las que no exista un alumno que a su vez sea profesor que no esté matriculado.

```
select idAsignatura  
from asignatura A  
where not exists
```

```
((select dni from alumno  
  intersect  
  select dni from profesor)  
except
```

Alumnos que son profesores

```
select A1.dni  
from alumno A1 join alumnosasignatura AA on  
  A1.IdAlumno=AA.IdAlumno  
where A.idasignatura = AA.idasignatura -- correlación)
```

Alumnos matriculados en la
asignatura (correlacionada)

Resolviendo divisiones

- ◆ Nombre y apellidos de los empleados que dedican 5 o más horas a todos los proyectos de los departamentos con 3 empleados.

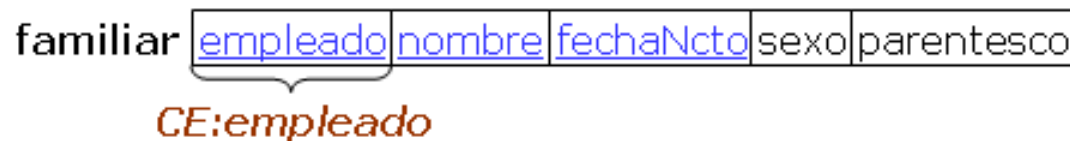
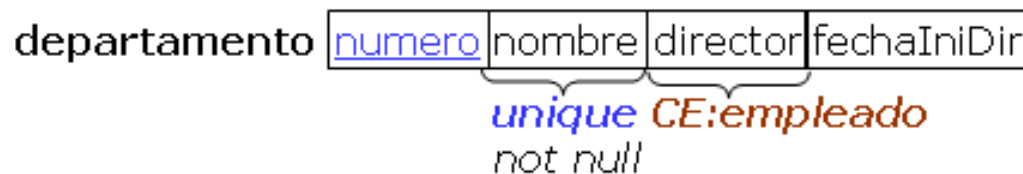
División!!

Nombre y apellidos de los empleados para los que no existe un proyecto de un departamento con 3 empleados a los que no le dedique 5 o más horas.

Resolviendo divisiones

nombre y apellidos de los empleados para los que no existe un proyecto de un departamento con 3 empleados a los que no le dedique 5 o más horas.

ESQUEMA BD EMPRESA



unique = clave candidata

Resolviendo divisiones

Nombre y apellidos de los empleados para los que no existe un proyecto de un departamento con 3 empleados a los que no le dedique 5 o más horas.

```
select e.Nombre, e.Apellidos  
from empleado e  
where not exists
```

```
(select p.numero  
from proyecto p  
where p.Depto in (select e1.Depto  
                  from empleado e1  
                  group by e1.Depto  
                  having count(e1.NSS)=3)
```

Proyectos de los departamentos
con 3 empleados

```
and p.numero not in(  
select t.Proyecto  
from trabaja t  
where t.Empleado=e.NSS -- correlación  
and t.horas >= 5));
```

Proyectos en los que el empleado
(correlación) le dedica más de 5
horas

Resolviendo divisiones

90. ID de los alumnos con todas las asignaturas de primer curso de Empresariales.

151. Listado de las asignaturas con todos los alumnos de la BD

persona dni nombre apellido ciudad direccionCalle direccionNum telefono fechaNacimiento varon

alumno idAlumno dni
unique
CE:persona
 not null

profesor idProfesor dni
unique
CE:persona
 not null

titulacion idTitulacion nombre
unique
 not null

asignatura idAsignatura nombre idTitulacion creditos cuatrimestre costeBasico idProfesor curso
unique not null *CE:titulacion* not null not null *CE:profesor* not null

alumnoAsignatura idAlumno idAsignatura numeroMatricula
CE:alumno CE:asignatura *Nº de veces que se ha matriculado*

Ejercicios consultas universidad

92. DNI de la alumna *Malena Folgado* y nombre de la titulación donde realizó más matriculas en sus asignaturas. La 1ª vez que se matricula de una asignatura *NumeroMatrícula* vale 1, la 2ª vale 2, etc. Atención: puede haber varias alumnas con igual nombre y apellido.

Solución 92

```
select P.dni, T.nombre
from persona P join
      alumno A on p.dni= A.dni join
      alumnoAsignatura AA on A.idAlumno=AA.idAlumno join
      asignatura Asg on AA.idAsignatura=Asg.idAsignatura join
      titulacion T on Asg.idTitulacion=T.idTitulacion
where P.nombre='Malena' and P.apellido='Folgado'
group by P.dni, T.nombre
having sum(AA.numeroMatricula) >= all
```

Es clave (unique), si no, habría que añadir idTitulacion

```
(select sum(AA2.numeroMatricula)
from alumnoAsignatura AA2 join
      asignatura Asg2 on AA2.idAsignatura=Asg2.idAsignatura
where AA2.idAlumno=A.idAlumno
group by Asg2.idTitulacion
)
```

No es lo mismo poner **nombre="Malena" and apellido="Folgado"** ya que no son clave en persona

cicios consultas universidad

Id de profesores que dan alguna asignatura con más de tres alumnos de su ciudad

```
select Pr.idProfesor
from persona as PP join
profesor as Pr on PP.dni=Pr.dni join
asignatura as A on Pr.idProfesor=A.idProfesor
where A.idAsignatura in
```

Asignaturas con más de tres alumnos de la ciudad del profesor PP

```
( select Asg.idAsignatura
  from asignatura as Asg join
      alumnoAsignatura as AA
      on Asg.idAsignatua=AA.idAsignatura join
      alumno as Al
      on AA.idAlumno=Al.idAlumno join
      persona PA on Al.dni=PA.dni
  where PA.ciudad=PP.ciudad -- correlación
  group by Asg.idAsignatura
  having count(*)>3
);
```

Cuestiones sueltas

left/right/full [outer] join (ejemplo)

PERSONA

<u>ID</u>	NOMBRE
A1	Juan
A2	María
A3	Pedro
A4	Juan

COMPRA

<u>PERSONA</u>	PRODUCTO
A1	P2
A2	P1
A2	P2

Pensar ejemplos en los que se unan estas tablas y sea necesario utilizar outer join

PRODUCTO

<u>ID</u>	NOMBRE	PRECIO
P1	Patatas	2
P2	Naranjas	2.5
P3	Sal	1

Group by de MySQL (1/2)

campos escondidos

- ◆ Admite campos (sin función) en **select** que no estén en **group by**

```
select A.idAsignatura, A.nombre, count(*)  
from Asignatura A join  
    AlumnoAsignatura AA on  
        A.idAsignatura=AA.idAsignatura  
group by A.idAsignatura
```

Sin problema: sólo hay un nombre por idAsignatura

Group by de MySQL (2/2)

campos escondidos

◆ ¿Qué pasa aquí?:

```
select A.idTitulacion, A.nombre, count(*)  
from Asignatura A  
group by A.idTitulacion
```

Resultados impredecibles

Access:

¿cómo hacer count(distinct ...)?

◆ Opción 1:

Cuántos costeBasico diferentes hay en las asignaturas:

```
select count(*)  
from ( select distinct costeBasico  
        from Asignatura  
      )
```

Equivalencias

$A = \text{any}(\text{subconsulta}) \Leftrightarrow A \text{ in } (\text{subconsulta})$

$A \neq \text{any}(\text{subconsulta}) \Leftrightarrow A \text{ not in } (\text{subconsulta})?$

Contraejemplos:

♦ $7 \neq \text{any}(7,4)$: devuelve **CIERTO** ($7 \neq 4$).

♦ $7 \text{ not in } (7,4)$: devuelve **FALSO**.

♦ $7 \neq \text{any}(\emptyset)$: devuelve **FALSO**.

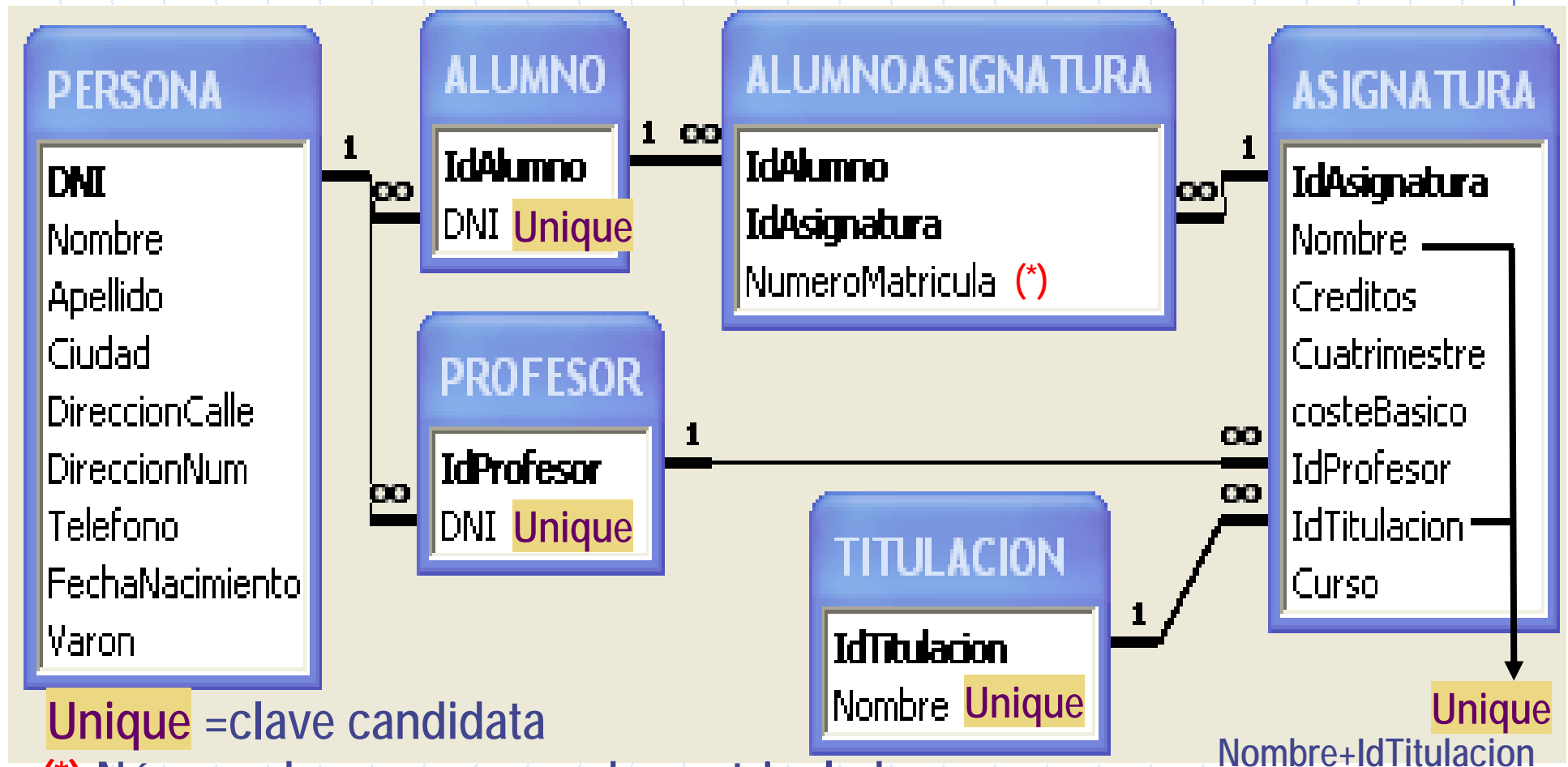
♦ $7 \text{ not in } (\emptyset)$: devuelve **CIERTO**.

Ejercicios

BD

Universidad

Esquema de la BD universidad



(*) Número de veces que se ha matriculado

1-8 9-15 16-19 20-24 25-29 30-34 35-39 40-42 43-45 46-48 49-49
 50-54 55-59 60-62 63-66 67-70 71-74 75-78 79-83 84-86 87-89 90-91 92...

Estado de la BD universidad (1/2)

	DNI	Nombre	Apellido	Ciudad	DireccionCalle	DireccionNum	Telefono	FechaNacimiento	Varon
+	16161616A	Luis	Ramírez	Haro	Pez	34	941111111	01/01/1969	1
+	17171717A	Laura	Beltran	Madrid	Gran Vía	23	912121212	08/08/1974	0
+	18181818A	Pepe	Pérez	Madrid	Percebe	13	913131313	02/02/1980	1
+	19191919A	Juan	Sánchez	Bilbao	Melancolía	7	944141414	03/03/1966	1
+	20202020A	Luis	Jiménez	Nájera	Cigüeña	15	941151515	03/03/1979	1
+	21212121A	Rosa	García	Haro	Alegría	16	941161616	04/04/1978	0
+	23232323A	Jorge	Sáenz	Logroño	Luis de Ulloa	17	941171717	09/09/1978	1
+	24242424A	María	Gutierrez	Logroño	Avda. la Paz	18	941181818	10/10/1964	0
+	25252525A	Rosario	Díaz	Logroño	Percebe	19	941191919	11/11/1971	0
+	26262626A	Elena	González	Logroño	Percebe	20	941202020	05/05/1975	0

	IdAsignatura	Nombre	Creditos	Cuatrimestre	costeBasico	IdProfesor	IdTitulacion	Curso
▶	+ 000115	Seguridad Vial	4,5	1	30	P204		
	+ 130113	Programación I	9	1	60	P101	130110	1
	+ 130122	Análisis II	9	2	60	P203	130110	2
	+ 150212	Química Física	4,5	2	70	P304	150210	1
	+ 160002	Contabilidad	6	1	70	P117	160000	1

Estado de la BD universidad (2/2)

PROFESOR : Tabla			ALUMNO : Tabla		
	IdProfesor	DNI		IdAlumno	DNI
+	P101	19191919A	▶	A010101	21212121A
+	P117	25252525A		A020202	18181818A
+	P203	23232323A		A030303	20202020A
+	P204	26262626A		A040404	26262626A
+	P304	24242424A		A121212	16161616A
				A131313	17171717A
			*		
registro: 1			Registro: 1		

TITULACION : Tabla			ALUMNOASIGNATURA : Tabla			
	IdTitulacion	Nombre		IdAlumno	IdAsignatura	NumeroMatricul
+	130110	Matemáticas	▶	A010101	150212	
+	150210	Químicas		A020202	130113	
+	160000	Empresariales		A020202	150212	
				A030303	130113	
				A030303	130122	
				A030303	150212	
				A040404	130122	
				A121212	000115	
				A131313	160002	
registro: 1						

Ejercicios consultas (1/23)

1. Nombre y apellidos de todas las personas
2. Códigos, nombres y créditos de las asignaturas.
3. Listado de todas las personas de la BD.
4. Posibilidades de créditos de las asignaturas (sin repeticiones).
5. Relación de los posibles costes básicos de asignaturas en cada curso.
6. Suma y media del coste básico incrementado el 10% de las asignaturas
7. ¿Cuántas asignaturas hay?
8. Coste de la asignatura más cara.

Ejercicios consultas (2/23)

9. Coste de la asignatura más barata.



10. Créditos y coste máximo, mínimo y medio de las asignaturas. Poner nombres de columna apropiados

11. ¿Qué posibilidades de créditos de asignatura hay?



12. ¿Cuántos cursos hay?



13. ¿Cuántas ciudades y nombres de personas distintos hay?




14. Nombre y créditos de las asignaturas de primer cuatrimestre.

15. Nombre y coste básico de las asignaturas de más de 4,5 créditos.

Ejercicios consultas (3/23)

- 
- 16. Nombre de las asignaturas con coste básico entre 25 y 35€ junto a las que no tienen todavía coste básico ordenadas por coste básico (con y sin BETWEEN).
 - 17. Id de los alumnos (sin repeticiones) matriculados en alguna de las siguientes asignaturas: 1502, 1301, 2014 o 1590 (con y sin IN).
 -  18. Nombre y apellido de los alumnos que tengan cualquier asignatura menos la 150 y la 130.
 - 19. Nombre de las asignaturas del segundo cuatrimestre que no sean de 6 créditos.

Ejercicios consultas (4/23)

20. Mostrar el nombre de las asignaturas cuyo coste por crédito sea mayor de 8 €.
21. Nombre y número de horas de todas las asignaturas (1 crédito = 10 h.).
-  22. Mostrar el nombre y apellido de las personas cuyo apellido comience por la letra G.
23. Datos de las personas que hayan nacido en 1978.
24. Mostrar el nombre de las personas para las que se desconoce la fecha de nacimiento.

Ejercicios consultas (5/23)

25. Mostrar las asignaturas que no pertenecen a ninguna titulación.


26. Listado del nombre completo de las personas, sus teléfonos y sus direcciones, llamando a la columna del nombre "Nombre Completo" y a la de direcciones "Dirección".

27. Nombre y edad de las personas con más de 20 años.

28. DNI y nombre de las personas que cumplan mañana años

29. Listado de personas ordenadas por apellidos y nombre.

Ejercicios consultas (6/23)

- 30. Listado de personas mayores de 25 años ordenadas por apellidos y nombre.
- 31. Listado que muestre las asignaturas con su coste por crédito ordenadas por su coste por crédito.
- 32. Listado de nombres completos de todos los alumnos
-  33. Listado de alumnos matriculados que viven en La Rioja (prefijo 941)
- 34. Listado de asignaturas impartidas por profesores de Logroño

Ejercicios consultas (7/23)

35. Profesores que además son alumnos.

36. Nombres completos de los profesores que además son alumnos.

37. Nombres completos de los alumnos que cursan la asignatura de Seguridad Vial.

38. Nombres completos de los alumnos que cursan alguna asignatura de 2º cuatrimestre.

39. Nombres de los profesores que imparten por lo menos una asignatura.

Ejercicios consultas (8/23)


40. Suma de los créditos de las asignaturas de Matemáticas

41. Número de asignaturas de la titulación de Matemáticas



42. Id de las titulaciones en las que está matriculado el alumno con DNI 18181818A (interpretando que para que un alumno esté matriculado en una titulación basta con que esté matriculado en alguna de sus asignaturas)

Ejercicios consultas (9/23)

- 
- 43. ¿Cuál sería el coste global de cursar la titulación de Matemáticas si el coste de cada asignatura fuera incrementado un 7%?
 - 44. Profesores que son alumnos indicando la asignatura o asignaturas que cursan.
 - 45. Nombre de las titulaciones en las que imparte docencia cada profesor, junto con el nombre de cada profesor.

Ejercicios consultas (10/23)

46. Listado ordenado por nombre de titulación y nombre de asignatura de todas las asignaturas mostrando además la titulación, nombre de asignatura y nombre del profesor.
47. Nombres completos de alumnos matriculados en alguna asignatura mostrando cuál es la asignatura y la titulación a la que pertenecen las asignaturas.
48. Listado que muestre, por cada asignatura, los nombres completos de los alumnos matriculados.

Ejercicios consultas (11/23)

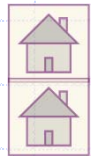
49. Supóngase que disponemos de la siguiente tabla, que clasifica la duración de las asignaturas en función de su número de créditos:

DURACION ASIGNATURA		
Duracion	CreditosMin	CreditosMax
Corta	0	4.5
Media	6	7.5
Larga	9	100

Obtener un listado de las asignaturas junto con la calificación de su duración según la tabla anterior.

Ejercicios consultas (12/23)

50. Asignaturas de la titulación de Matemáticas.



51. Asignaturas con más créditos que la asignatura de libre elección (sin titulación) "Seguridad Vial".

52. Listado de alumnos que son mayores que **algún** profesor, indicando cada alumno y cada profesor.

53. Listado de alumnos que son mayores que **alguno de sus** profesores, indicando cada alumno y cada profesor.

54. Cuál es el coste de la matrícula de cada titulación.

Ejercicios consultas (13/23)



55. Cuántos alumnos hay matriculados en cada asignatura.

56. Cuánto paga cada alumno por su matrícula.

57. Estadísticas (número y media de edad) de personas por ciudades y sexos.

58. Lo mismo que la anterior pero por mes de nacimiento, y sólo para los alumnos.



59. Identificador de cada titulación junto al coste básico medio de sus asignaturas para aquellas titulaciones cuyo coste total de matrícula sea mayor de 600 €.

Ejercicios consultas (14/23)

60. Para las titulaciones con más de 20 asignaturas, visualiza el código de titulación, el mayor número de créditos en una asignatura, la media y la suma de los créditos.
61. Nombre y apellidos de los alumnos matriculados al menos en dos asignaturas. Ordena el resultado alfabéticamente.
62. Nombre y apellidos de alumnos cuya matrícula en asignaturas de la titulación de 'Matemáticas' haya supuesto un coste superior a 500 €.



Ejercicios consultas (15/23)

- 63. Clasifica las titulaciones por orden descendente en función del total de créditos necesarios para cursarlas.
- 64. Edad del alumno más viejo de cada asignatura.
- 65. Lista de gente ordenada por el mes en el que cumplen los años.
- 66. Identificador de los alumnos matriculados en las asignaturas 150212 y 130113 (en ambas a la vez).



Ejercicios consultas (16/23)



67. Nombre de las asignaturas de la titulación 130110 cuyo coste básico sobrepase el coste básico promedio por asignatura en dicha titulación.



68. Lista de asignaturas en las que no se ha matriculado nadie.




69. Nombre de las titulaciones que tengan al menos una asignatura de más de 4,5 créditos.



70. Asignaturas con más créditos que alguna de las asignaturas de Matemáticas.

Ejercicios consultas (17/23)

71. Listado de personas de mayor edad que todas las personas de Haro.
72. Listado de asignaturas que pertenecen al mismo cuatrimestre que la asignatura de menor coste y que tienen unos créditos superiores a la media de las asignaturas.
73. Listado de asignaturas cuyo coste es superior al coste medio de las asignaturas que no pertenecen a ninguna titulación.
-  74. Listado de personas que viven en alguna de las ciudades donde viven más personas de la BD.

Ejercicios consultas (18/23)

- 75. Listado de alumnos que nacieron antes que el profesor más joven.
- 76. Asignaturas de la titulación 130110 en las que no está matriculado el alumno A020202.
- 77. Listado de alumnos cuya edad supera la media de la edad de los profesores.
- 78. Listado de alumnos que asisten a alguna asignatura de las impartidas por profesores varones de Logroño y cuyo número de compañeros de asignatura sea menor que 10.

Ejercicios consultas (19/23)

- 79. Listado de asignaturas no cursadas por alumnos de Logroño.
- 80. Para cada asignatura con más de dos matriculados encuentra el número de alumnos matriculados en ella de más de 25 años.
- 81. Listado de ciudades en las que vive algún profesor pero no un alumno (y lo contrario).
- 82. Listado de ciudades en las que vive algún profesor y también algún alumno.
- 83. Listado de alumnos (ids) no matriculados en asignaturas de matemáticas.



Ejercicios consultas (20/23)

84. Listar los ids de los alumnos para los que se cumpla que la suma de créditos de las asignaturas en las que está matriculado sea superior a la suma de créditos de todas las asignaturas de Empresariales.
85. Identificadores de los alumnos más viejos de cada titulación.
86. Listado de DNI de personas relacionadas con la asignatura 'Seguridad Vial', bien por ser profesor de la misma o por estar matriculado en la misma.

Ejercicios consultas (21/23)

87. Listado de nombres completos de personas relacionadas con la asignatura 'Seguridad Vial', bien por ser profesor de la misma o por estar matriculado en la misma.
88. Escribir una consulta que permita verificar si todas las personas de la BD son o bien profesores o bien alumnos, mostrando los DNIs de las personas que no lo cumplen
89. Mostrar el ID de los alumnos matriculados en todas las asignaturas de primer curso de la titulación 130110

Ejercicios consultas (22/23)



90. Mostrar el ID de los alumnos matriculados en todas las asignaturas de primer curso de Empresariales.
91. Estadísticas por titulación (mostrando el nombre de la titulación): coste básico máximo, mínimo y medio y créditos medios de sus asignaturas. Incluye en la lista las asignaturas que no tienen asociada ninguna titulación, mostrando el texto "Asignatura de campus" en el nombre de la titulación.

Ejercicios consultas (23/23)

92. DNI de la alumna *Malena Folgado* y nombre de la titulación donde realizó más matriculas en sus asignaturas. La 1ª vez que se matricula de una asignatura *NumeroMatrícula* vale 1, la 2ª vale 2, etc. Atención: puede haber varias alumnas con igual nombre y apellido.
93. ¿En qué titulación hay matriculados más alumnos por primera vez (todas las asignaturas de esa titulación con *númeroMatrícula*=1), en matemáticas o en ITIG?