

SQL con SQLite

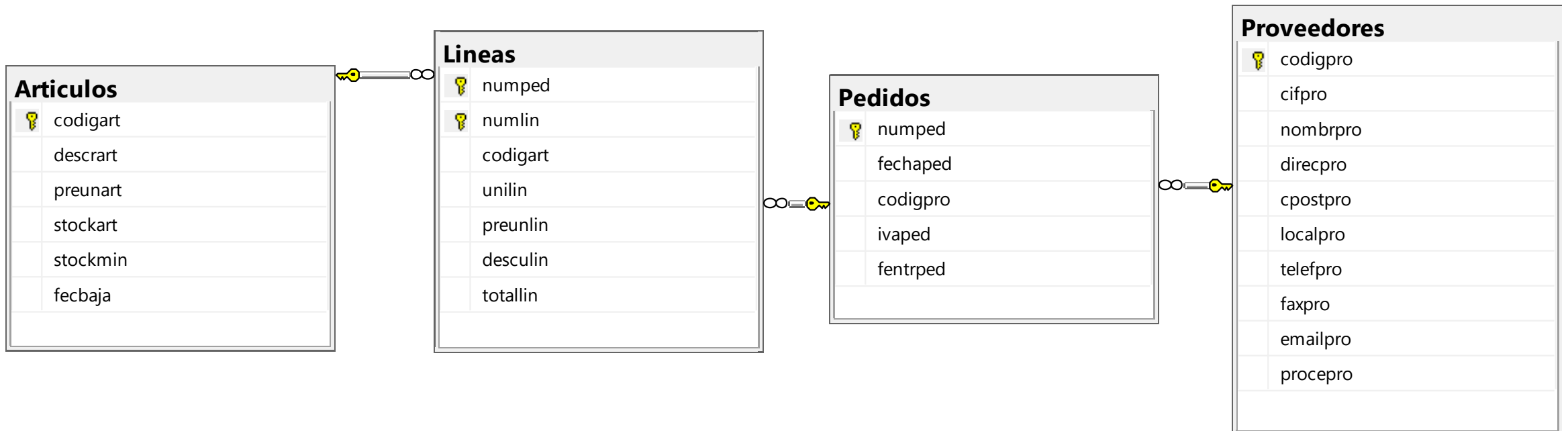
Agregaciones, Joins y Subconsultas

Máster en Data Science

M1967 - Modelos de Datos y Sistemas de Información 2018-2019

Agrupaciones, Joins y Subconsultas

- En este tema, veremos como realizar consultas con agregaciones, joins y subconsultas
- Trabajaremos con la siguiente base de datos, cuyos ficheros .sql para su creación e inserción de datos se encuentran colgados en Moodle:



SECCIÓN 1: JOINS

Consultas en SQL: JOINS

- En el modelo relacional, es frecuente realizar consultas que retornen datos de varias tablas.
- Para ello, se usa la clausula JOIN:
 - Ejemplo 1: devolver los datos de los proveedores junto con los datos de todos sus pedidos:

Une las filas de las dos tablas

```
SELECT pr.*, pe.*  
FROM Proveedores pr JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro;
```

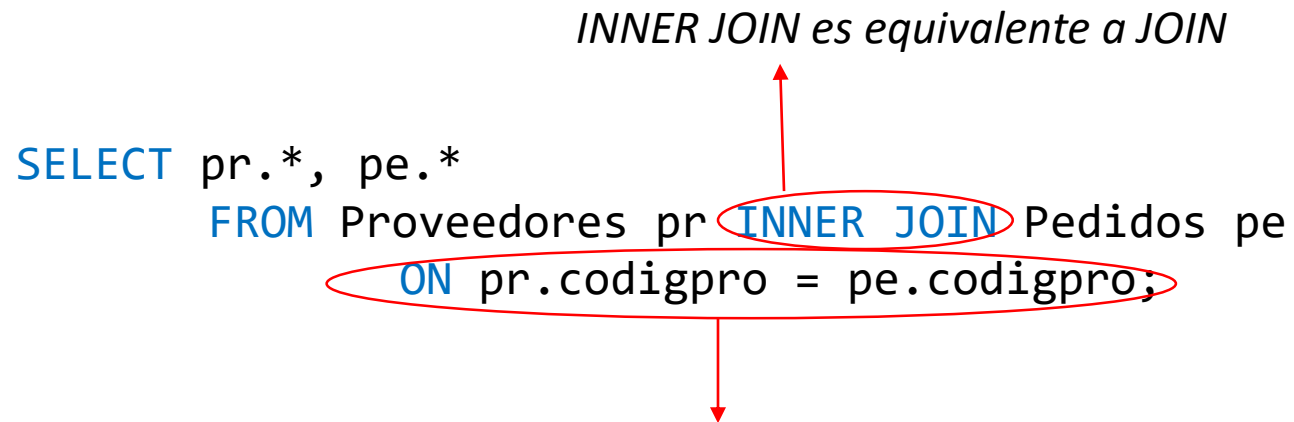
Condición de unión: cada fila de proveedores sólo se une con las filas de pedidos que sean del mismo proveedor. Esto es, al lado del proveedor 'P001' solo aparecerán los datos de sus pedidos. Si un proveedor no tiene pedidos, no aparece.

Consultas en SQL: JOINS

- En el modelo relacional, es frecuente realizar consultas que retornen datos de varias tablas.
- Para ello, se usa la clausula JOIN (equivalente a INNER JOIN):
 - Ejemplo 1: devolver los datos de los proveedores junto con los datos de todos sus pedidos:

INNER JOIN es equivalente a JOIN

```
SELECT pr.*, pe.*  
FROM Proveedores pr INNER JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro;
```



Condición de unión: cada fila de proveedores sólo se une con las filas de pedidos que sean del mismo proveedor. Esto es, al lado del proveedor 'P001' solo aparecerán los datos de sus pedidos. Si un proveedor no tiene pedidos, no aparece.

Consultas en SQL: JOINS

- En el modelo relacional, es frecuente realizar consultas que retornen datos de varias tablas.
- Para ello, se usa la clausula JOIN:
 - Ejemplo 1: devolver los datos de los proveedores junto con los datos de todos sus pedidos:

```
SELECT pr.*, pe.*  
FROM Proveedores pr JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro;
```

	codigpro	cifpro	nombrpro	direcpro	cpostpro	localpro	telepro	faxpro	emailpro	procepro	numped	fechaped	codigpro	ivaped	fentrped
1	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	1	2010-05-22	P001	21	2010-06-16
2	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bilbao	(34) 947 865 413	(34) 947 865 434	mailto:zarana@yahoo.es	UE	2	2010-06-10	P002	18	2010-08-15
3	P003	B28373212	Gras Leon, Luz	Pez 14, 5C dcha.	28119	Madrid	(34) 916 677 829	(34) 916 677 889	NULL	UE	3	2010-10-15	P003	18	2010-12-15
4	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	4	2010-08-13	P001	18	2010-09-10
5	P004	B85392314	Gil Laso, Luis	Uria 18, 2F	85223	Oviedo	(34) 952 345 6632	(34) 952 345 678	NULL	UE	5	2010-11-13	P004	21	2010-11-13

Consultas en SQL: JOINS

- En el modelo relacional, es frecuente realizar consultas que retornen datos de varias tablas.
- Para ello, se usa la clausula JOIN:
 - Ejemplo 2: devolver los datos de los artículos junto con las líneas de pedido en las que aparecen.

```
SELECT a.*, l.*  
FROM Articulos a JOIN Lineas l  
ON a.codigart = l.codigart  
ORDER BY a.codigart;
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	1	0001	1	220	0
2	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	1	0002	3	120	2
3	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	3	0002	5	120	0
4	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3	1	0002	1	110	0
5	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4	1	0002	4	120	0
6	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	1	2	0003	2	295	0
7	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	2	2	0003	2	300	3
8	0004	ARCHIVADOR MOD. TR	180	3	1	NULL	4	2	0004	10	180	0

Consultas en SQL: JOINS

- ¿Y qué sucede si no incluimos la clausula 'ON'?
- Pues que se unen todas las filas de la primera tabla con todas las filas de la segunda... (CROSS JOIN)

```
SELECT pr.*, pe.*  
FROM Proveedores pr JOIN Pedidos pe
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	1	0001	1	220	0
2	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	2	0003	2	295	0
3	0001	MESA OFICINA 90x1,80	225	100	1	NULL	2	1	0002	3	120	2
4	0001	MESA OFICINA 90x1,80	225	100	1	NULL	2	2	0003	2	300	3
5	0001	MESA OFICINA 90x1,80	225	100	1	NULL	2	3	0002	5	120	0
6	0001	MESA OFICINA 90x1,80	225	100	1	NULL	3	1	0002	1	110	0
7	0001	MESA OFICINA 90x1,80	225	100	1	NULL	4	1	0002	4	120	0
8	0001	MESA OFICINA 90x1,80	225	100	1	NULL	4	2	0004	10	180	0
9	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	1	1	0001	1	220	0
10	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	1	2	0003	2	295	0
11	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	1	0002	3	120	2
12	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	2	0003	2	300	3
13	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	3	0002	5	120	0
14	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3	1	0002	1	110	0
15	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4	1	0002	4	120	0
16	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4	2	0004	10	180	0

¡Siempre hay que usar la condición de unión en el JOIN!

Consultas en SQL: JOINS

- Al igual que en las consultas sobre una sola tabla, podemos introducir condiciones en WHERE.
 - Ejemplo 1: devolver los datos de los artículos junto con las líneas de pedido en las que aparecen, siempre que el precio unitario del artículo sea mayor que el de la línea.

```
SELECT a.*, l.*  
FROM Articulos a JOIN Lineas l  
ON a.codigart = l.codigart  
WHERE a.preunart > l.preunlin  
ORDER BY a.codigart;
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	1	0001	1	220	0
2	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3	1	0002	1	110	0
3	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	1	2	0003	2	295	0

Consultas en SQL: JOINS

- Al igual que en las consultas sobre una sola tabla, podemos introducir condiciones en WHERE.
 - Ejemplo 2: devolver los datos de los artículos junto con las líneas de pedido en las que aparecen, mostrando sólo aquellos casos en los que el artículo no se haya dado de baja y en los que no se haya aplicado ningún descuento.

```
SELECT a.*, l.*  
FROM Articulos a JOIN Lineas l  
ON a.codigart = l.codigart  
WHERE a.fecbaja IS null AND l.desculin = 0  
ORDER BY a.codigart;
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1 1		0001	1	220	0
2	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2 3		0002	5	120	0
3	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3 1		0002	1	110	0
4	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4 1		0002	4	120	0
5	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	1 2		0003	2	295	0
6	0004	ARCHIVADOR MOD. TR	180	3	1	NULL	4 2		0004	10	180	0

Consultas en SQL: JOINS

- Una alternativa a la clausula ON es usar USING o NATURAL JOIN.
¡Cuidado!, solo funciona si la PK de una tabla y la FK de la otra se llaman exactamente igual:
 - Devolver los datos de los proveedores junto con los datos de todos sus pedidos:

```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      NATURAL JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro;
```

```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      JOIN Pedidos pe  
USING(codigpro);
```

```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro;
```

	codigpro	cifpro	nombrpro	direcpro	cpostpro	localpro	telepro	faxpro	emailpro	procepro	numped	fechaped	codigpro	ivaped	fentrped
1	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	1	2010-05-22	P001	21	2010-06-16
2	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bilbao	(34) 947 865 413	(34) 947 865 434	mailto:zarana@yahoo.es	UE	2	2010-06-10	P002	18	2010-08-15
3	P003	B28373212	Gras Leon, Luz	Pez 14, 5C dcha.	28119	Madrid	(34) 916 677 829	(34) 916 677 889	NULL	UE	3	2010-10-15	P003	18	2010-12-15
4	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	4	2010-08-13	P001	18	2010-09-10
5	P004	B85392314	Gil Laso, Luis	Uria 18, 2F	85223	Oviedo	(34) 952 345 6632	(34) 952 345 678	NULL	UE	5	2010-11-13	P004	21	2010-11-13

Consultas en SQL: JOINS

- Otra alternativa es poner la condición de unión en el WHERE, junto con el resto de condiciones. En esta asignatura, por homogeneidad, se recomienda usar el ON:
 - Devolver los datos de los proveedores junto con los datos de todos sus pedidos:


```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro;
```

```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      JOIN Pedidos pe  
WHERE pr.codigpro = pe.codigpro;
```

	codigpro	cifpro	nombrpro	direcpro	cpostpro	localpro	telefpro	faxpro	emailpro	procepro	numped	fechaped	codigpro	ivaped	fentrped
1	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	1	2010-05-22	P001	21	2010-06-16
2	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bilbao	(34) 947 865 413	(34) 947 865 434	mailto:zarana@yahoo.es	UE	2	2010-06-10	P002	18	2010-08-15
3	P003	B28373212	Gras Leon, Luz	Pez 14, 5C dcha.	28119	Madrid	(34) 916 677 829	(34) 916 677 889	NULL	UE	3	2010-10-15	P003	18	2010-12-15
4	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	4	2010-08-13	P001	18	2010-09-10
5	P004	B85392314	Gil Laso, Luis	Uria 18, 2F	85223	Oviedo	(34) 952 345 6632	(34) 952 345 678	NULL	UE	5	2010-11-13	P004	21	2010-11-13

Consultas en SQL: JOINS

- ¡Cuidado con la ausencia de alias!
 - La siguiente consulta devolvería un error, tratando de retornar los pedidos del proveedor 'P001', junto con los datos del propio proveedor:

Status
 [15:30:46] Error while executing SQL query on database 'bdTema2': ambiguous column name: codigpro

```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro  
WHERE codigpro = 'P001';
```



Hay dos 'codigpro', uno en cada tabla... ¿a cuál se refiere?

Consultas en SQL: JOINS

- ¡Cuidado con la ausencia de alias!
 - La siguiente consulta sería correcta, al darle un alias al campo cuyo nombre se repite en las dos tablas:

```
SELECT pr.*, pe.*  
FROM Proveedores pr  
      JOIN Pedidos pe  
ON pr.codigpro = pe.codigpro  
WHERE pr.codigpro = 'P001';
```

	codigpro	cifpro	nombrpro	direcpro	cpostpro	localpro	telefpro	faxpro	emailpro	procepro	numped	fechaped	codigpro	ivaped	fentrpel
1	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	1	2010-05-22	P001	21	2010-06-16
2	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Santander	(34) 942 223 345	(34) 942 223 344	mailto:baupi@eresmas.es	UE	4	2010-08-13	P001	18	2010-09-10

Consultas en SQL: JOINS

- Podemos unir más de dos tablas en el FROM.
 - Ejemplo 1: devolver los datos de los artículos junto con las líneas de pedido en las que aparecen, y junto con los datos de los pedidos

```
SELECT a.*, l.*  
      FROM Articulos a  
      JOIN Lineas l ON a.codigart = l.codigart  
      JOIN Pedidos pe ON pe.numped = l.numped;
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin	numped:	fechaped	codigpro	ivaped	fentrped
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	1	0001	1	220	0	1	2010-05-22	P001	21	2010-06-16
2	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	1	2	0003	2	295	0	1	2010-05-22	P001	21	2010-06-16
3	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	1	0002	3	120	2	2	2010-06-10	P002	18	2010-08-15
4	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	2	2	0003	2	300	3	2	2010-06-10	P002	18	2010-08-15
5	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	3	0002	5	120	0	2	2010-06-10	P002	18	2010-08-15
6	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3	1	0002	1	110	0	3	2010-10-15	P003	18	2010-12-15
7	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4	1	0002	4	120	0	4	2010-08-13	P001	18	2010-09-10
8	0004	ARCHIVADOR MOD. TR	180	3	1	NULL	4	2	0004	10	180	0	4	2010-08-13	P001	18	2010-09-10

Consultas en SQL: JOINS

- Podemos unir más de dos tablas en el FROM.
 - Ejemplo 2: devolver los datos de los artículos junto con las líneas de pedido en las que aparecen, junto con los datos de los pedidos y de los proveedores

```
SELECT a.*, l.*, pe.*, pr.*
FROM Articulos a
JOIN Lineas l ON a.codigart = l.codigart
JOIN Pedidos pe ON pe.numped = l.numped
JOIN Proveedores pr ON pe.codigpro = pr.codigpro
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin	numped:	fechaped	codigpro	ivaped	fentrp	codigpro	cifpro	nombpro	direcpro	cpostpro	loc
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	1	0001	1	220	0	1	2010-05-22	P001	21	2010-06-16	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Sa
2	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	1	2	0003	2	295	0	1	2010-05-22	P001	21	2010-06-16	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Sa
3	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	1	0002	3	120	2	2	2010-06-10	P002	18	2010-08-15	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bill
4	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	2	2	0003	2	300	3	2	2010-06-10	P002	18	2010-08-15	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bill
5	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	3	0002	5	120	0	2	2010-06-10	P002	18	2010-08-15	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bill
6	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3	1	0002	1	110	0	3	2010-10-15	P003	18	2010-12-15	P003	B28373212	Gras Leon, Luz	Pez 14, 5C dcha.	28119	Ma
7	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4	1	0002	4	120	0	4	2010-08-13	P001	18	2010-09-10	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Sa
8	0004	ARCHIVADOR MOD. TR	180	3	1	NULL	4	2	0004	10	180	0	4	2010-08-13	P001	18	2010-09-10	P001	A39184215	Bau Pi, Pablo	Alta 3, 2D	39390	Sa

Consultas en SQL: JOINS

- Podemos unir más de dos tablas en el FROM.
 - Ejemplo 3: devolver los datos de los artículos junto con las líneas de pedido en las que aparecen, junto con los datos de los pedidos y de los proveedores. Filtrar devolviendo solamente los pedidos del proveedor 'P002' y cuyos artículos tengan un precio mayor a 200:

```
SELECT a.*, l.*, pe.*, pr.*  
      FROM Articulos a  
           JOIN Lineas l ON a.codigart = l.codigart  
           JOIN Pedidos pe ON pe.numped = l.numped  
           JOIN Proveedores pr ON pe.codigpro = pr.codigpro  
      WHERE pr.codigpro = 'P002' AND a.preunart > 200
```

	codigart	descart	preunart	stockart	stockmin	fechaja	numped	numlin	codigart:	unilin	preunlin	desculin	numped:	fechaped	codigpro	ivaped	fentrped	codigpro	cifpro	nombrpro	direcpro	cpostpro	localpro	tele
1	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	2	2	0003	2	300	3	2	2010-06-10	P002	18	2010-08-15	P002	A48162311	Zar Luna, Ana	Ercilla 22, 1A	48002	Bilbao	(34)

Consultas en SQL: JOINS

- En SQLite, se puede usar LEFT JOIN, en vez de INNER JOIN (o JOIN), para devolver la unión de dos tablas bajo un criterio y, además, devolver las filas de la tabla de la izquierda que no tengan correspondencia con ninguna fila de la otra tabla.
 - Ejemplo 1: devolver los datos de los artículos junto con los datos de todos sus líneas. Devolver también los datos de los artículos que no tengan aparezcan en ninguna línea de pedido.

```
SELECT a.*, l.*  
FROM Articulos a LEFT JOIN Lineas l  
ON a.codigart = l.codigart;
```

	codigart	descart	preunart	stockart	stockmin	fecbaja	numped	numlin	codigart:	unilin	preunlin	desculin
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL	1	1	0001	1	220	0
2	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	1	0002	3	120	2
3	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	2	3	0002	5	120	0
4	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	3	1	0002	1	110	0
5	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL	4	1	0002	4	120	0
6	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	1	2	0003	2	295	0
7	0003	ARMARIO DIPLOMATIC	300	2	1	NULL	2	2	0003	2	300	3
8	0004	ARCHIVADOR MOD. TR	180	3	1	NULL	4	2	0004	10	180	0
9	0005	PANTALLA 19	110	2	1	2018-02-06	NULL	NULL	NULL	NULL	NULL	NULL


AHORA TOCA HACER EL EJERCICIO 7

SECCIÓN 2: AGREGACIONES

Consultas en SQL: Agregaciones

- Operadores de agregación. Sirven para agrupar los valores en base a un criterio. Por ejemplo: AVG (media), MAX (máximo), MIN (mínimo), COUNT (número de coincidencias)...


- Ejemplo 1: obtener el precio del artículo más caro



	maximo
1	300

```
SELECT MAX(a.preunart) maximo  
FROM Articulos a
```


- Ejemplo 2: obtener el pedido más antiguo



	minimo
1	2010-05-22

```
SELECT MIN(pe.fechaped) minimo  
FROM Pedidos pe
```


- Ejemplo 3: obtener el iva medio de los pedidos al proveedor 'P001'



	media
1	19.5

```
SELECT AVG(pe.ivaped) media  
FROM Pedidos pe WHERE pe.codigpro = 'P001'
```

- Ejemplo 4: número de proveedores en la base de datos



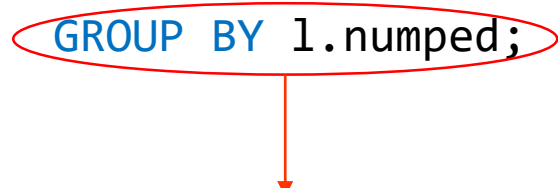
	numero
1	4

```
SELECT COUNT(pr.*) num  
FROM Proveedores pr
```

Consultas en SQL: Agregaciones

- Los operadores de agregación se pueden usar para realizar agrupaciones parciales, esto es, en base al valor de una columna. Para ello, se usa la cláusula GROUP BY:
 - Ejemplo 1: obtener el número de líneas por pedido

```
SELECT l.numped, COUNT(*) numLineas  
FROM Lineas l  
GROUP BY l.numped;
```



*En este caso, se indica que estamos agrupando según el valor de la columna numped.
Es decir, se ejecutará el COUNT para cada número de pedido.*

Consultas en SQL: Agregaciones

- Los operadores de agregación se pueden usar para realizar agrupaciones parciales, esto es, en base al valor de una columna. Para ello, se usa la cláusula GROUP BY:
 - Ejemplo 1: obtener el número de líneas por pedido

```
SELECT l.numped, COUNT(*) numLineas  
FROM Lineas l  
GROUP BY l.numped;
```

	numped	numLineas
1	1	2
2	2	3
3	3	1
4	4	2

Consultas en SQL: Agregaciones

- Los operadores de agregación se pueden usar para realizar agrupaciones parciales, esto es, en base al valor de una columna. Para ello, se usa la cláusula GROUP BY:
 - Ejemplo 2: obtener el número de unidades vendidas por artículo

```
SELECT l.codigart, SUM(l.unilin) numUnidades  
FROM Lineas l  
GROUP BY l.codigart;
```

	codigart	numUnidades
1	0001	1
2	0002	13
3	0003	4
4	0004	10

Consultas en SQL: Agregaciones

- Los operadores de agregación se pueden usar para realizar agrupaciones parciales, esto es, en base al valor de una columna. Para ello, se usa la cláusula GROUP BY:
 - Ejemplo 3: obtener la media, máximo y mínimo precio pagado por cada artículo

```
SELECT l.codigart, MAX(l.preunlin) maximo, MIN(l.preunlin) minimo,  
       AVG(l.preunlin) media  
FROM Lineas l  
GROUP BY l.codigart;
```

	codigart	maximo	minimo	media
1	0001	220	220	220
2	0002	120	110	117.5
3	0003	300	295	297.5
4	0004	180	180	180

Consultas en SQL: Agregaciones

- ¡CUIDADO! Si se quieren incluir otros campos fuera de la agregación, estos también tienen que aparecer en el GROUP BY:
 - Ejemplo 1: obtener el número de líneas por pedido, junto con la fecha del pedido

```
SELECT pe.numped, pe.fechaped, COUNT(*) numLineas  
FROM Lineas l JOIN Pedidos pe ON pe.numped = l.numped  
GROUP BY pe.numped, pe.fechaped;
```

	numped	fechaped	numLineas
1	1	2010-05-22	2
2	2	2010-06-10	3
3	3	2010-10-15	1
4	4	2010-08-13	2

Consultas en SQL: Agregaciones

- ¡CUIDADO! Si se quieren incluir otros campos fuera de la agregación, estos también tienen que aparecer en el GROUP BY:
 - Ejemplo 2: obtener la media, máximo y mínimo precio pagado por cada artículo, junto con la descripción del artículo, su stock y su fecha de baja

```
SELECT a.codigart, a.fecbaja, a.descrart, a.stockart,  
MAX(l.preunlin) maximo, MIN(l.preunlin) minimo, AVG(l.preunlin) media  
FROM Lineas l JOIN Articulos a ON a.codigart = l.codigart  
GROUP BY l.codigart, a.fecbaja, a.descrart, a.stockart;
```

	codigart	fecbaja	descrart	stockart	maximo	minimo	media
1	0001	NULL	MESA OFICINA 90x1,80	100	220	220	220
2	0002	NULL	SILLA ERGONOMIC MOD. MX	25	120	110	117.5
3	0003	NULL	ARMARIO DIPLOMATIC	2	300	295	297.5
4	0004	NULL	ARCHIVADOR MOD. TR	3	180	180	180

Consultas en SQL: Agregaciones

- Se pueden poner condiciones en el WHERE de la forma usual:
 - Ejemplo 1: obtener el número de líneas sin descuento por pedido

```
SELECT l.numped, COUNT(*) numLineas
FROM Lineas l
WHERE l.desculin =0
GROUP BY l.numped;
```

	numped	numLineas
1	1	2
2	2	1
3	3	1
4	4	2

Consultas en SQL: Agregaciones

- Se pueden poner condiciones en el WHERE de la forma usual:
 - Ejemplo 2: obtener el número de unidades vendidas por artículo, junto a la descripción del artículo. Filtrar solamente para aquellos artículos con stock superior al mínimo, y siempre que el precio unitario de venta fuese mayor a 200

```
SELECT l.codigart, a.descrart, SUM(l.unilin) numUnidades
FROM Lineas l INNER JOIN Articulo a ON a.codigart = l.codigart
WHERE a.stockart > a.stockmin AND l.preulin > 200
GROUP BY l.codigart, a.descrart;
```

	codigart	descrart	numUnidades
1	0001	MESA OFICINA 90x1,80	1
2	0003	ARMARIO DIPLOMATIC	4

Consultas en SQL: Agregaciones

- Si queremos poner condiciones sobre el resultado de la agregación, se utiliza la clausula HAVING:
 - Ejemplo 1: obtener el número de líneas por pedido, siempre que el número de líneas sea mayor a 1

```
SELECT l.numped, COUNT(*) numLineas
FROM Lineas l
GROUP BY l.numped
HAVING COUNT(*) > 1;
```

	numped	numLineas
1	1	2
2	2	3
3	4	2

Consultas en SQL: Agregaciones

- Si queremos poner condiciones sobre el resultado de la agregación, se utiliza la clausula HAVING:
 - Ejemplo 2: obtener el número de líneas sin descuento por pedido, siempre que el número de líneas que cumplan la restricción sea mayor a 2

```
SELECT l.numped, COUNT(*) numLineas
FROM Lineas l
WHERE l.desculin =0
GROUP BY l.numped
HAVING COUNT(*) > 1;
```

	numped	numLineas
1	1	2
2	4	2

Consultas en SQL: Agregaciones

- Si queremos poner condiciones sobre el resultado de la agregación, se utiliza la clausula HAVING:
 - Ejemplo 3: obtener el número de unidades vendidas por artículo, junto a la descripción del artículo. Filtrar solamente para aquellos artículos con stock superior al mínimo, y siempre que la suma de unidades vendidas sea mayor a 2

```
SELECT l.codigart, a.descrart, SUM(l.unilin) numUnidades
FROM Lineas l INNER JOIN Articulo a ON a.codigart = l.codigart
WHERE a.stockart > a.stockmin
GROUP BY l.codigart, a.descrart
HAVING SUM(l.unilin) > 2;
```

	codigart	descrart	numUnidades
1	0002	SILLA ERGONOMIC MOD. MX	13
2	0003	ARMARIO DIPLOMATIC	4
3	0004	ARCHIVADOR MOD. TR	10

Consultas en SQL: Agregaciones

- Las agrupaciones se pueden usar para ordenar
 - Ejemplo 1: obtener el número de líneas por pedido, siempre que el número de líneas sea mayor a 1, y ordenar por el número de líneas.

```
SELECT l.numped, COUNT(*) numLineas
FROM Lineas l
GROUP BY l.numped
ORDER BY COUNT(*)
HAVING COUNT(*) > 1;
```

	numped	numLineas
1	1	2
2	2	3
3	4	2

AHORA TOCA HACER EL EJERCICIO 8

SECCIÓN 3: SUBCONSULTAS

Consultas en SQL: Subconsultas

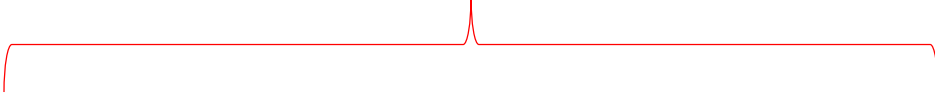
- Los SELECT se pueden embeber dentro de otros SELECT “de mayor” nivel.
- Esto aporta mayor potencia al lenguaje SQL, pudiendo hacer consultas complejas que, de otra forma, no podrían realizarse (o no, al menos de forma eficiente).
- Las subconsultas se pueden usar en la proyección (siempre que retornen un único valor escalar por fila), en el FROM (como si fuesen una tabla más) o en el WHERE.

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en el WHERE. Con el operador IN, podemos buscar coincidencias en una lista de valores devuelta por un SELECT.
 - Ejemplo 1: devolver los datos de todos los artículos que han sido comprados en alguna ocasión:

Esta subconsulta devuelve una lista con los códigos de todos los artículos que aparecen en la tabla Lineas

```
SELECT a.*  
FROM Articulos a  
WHERE a.codigart IN (SELECT l.codigart FROM Lineas l);
```



	codigart	descart	preunart	stockart	stockmin	fecbaja
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL
2	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL
3	0003	ARMARIO DIPLOMATIC	300	2	1	NULL
4	0004	ARCHIVADOR MOD. TR	180	3	1	NULL

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en el WHERE. Con el operador IN, podemos buscar coincidencias en una lista de valores devuelta por un SELECT.
 - Ejemplo 2: devolver los datos de todos los artículos que NO han sido comprados en alguna ocasión:

Esta subconsulta devuelve una lista con los códigos de todos los artículos que aparecen en la tabla Lineas

```
SELECT a.*  
FROM Articulos a  
WHERE a.codigart NOT IN (SELECT l.codigart FROM Lineas l);
```



	codigart	descart	preunart	stockart	stockmin	fecbaja
1	0005	PANTALLA 19	110	2	1	2018-02-06

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en el WHERE. El operador EXISTS devuelve true si hay al menos una fila que cumpla las condiciones, y false en caso contrario
 - Ejemplo 1: devolver los datos de todos los artículos han sido comprados en alguna ocasión (con EXISTS):

Por cada artículo, si hay alguna Línea que coincida con el codigart de dicho artículo, devolverá true. Sino, devolverá false.

```
SELECT a.*  
FROM Articulos a  
WHERE EXISTS (SELECT * FROM Lineas l WHERE l.codigart = a.codigart);
```

	codigart	descart	preunart	stockart	stockmin	fecbaja
1	0001	MESA OFICINA 90x1,80	225	100	1	NULL
2	0002	SILLA ERGONOMIC MOD. MX	120	25	1	NULL
3	0003	ARMARIO DIPLOMATIC	300	2	1	NULL
4	0004	ARCHIVADOR MOD. TR	180	3	1	NULL

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en el WHERE. El operador EXISTS devuelve true si hay al menos una fila que cumpla las condiciones, y false en caso contrario
 - Ejemplo 2: devolver los datos de todos los artículos que NO han sido comprados en alguna ocasión (con EXISTS):

Por cada artículo, si hay alguna Línea que coincida con el codigart de dicho artículo, devolverá true. Sino, devolverá false.

```
SELECT a.*  
FROM Articulos a  
WHERE NOT EXISTS (SELECT * FROM Lineas l WHERE l.codigart = a.codigart);
```

	codigart	descart	preunart	stockart	stockmin	fecbaja
1	0005	PANTALLA 19	110	2	1	2018-02-06

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en el WHERE. También se pueden comparar valores de columnas con resultados de agregaciones.
 - Ejemplo: devolver los datos de los pedidos que se hayan enviado con posterioridad a los otros pedidos (es decir, los datos de el/los último/s pedido/s en enviarse)

```
SELECT pe.*  
FROM Pedidos pe  
WHERE pe.fechaped =  
      (SELECT MAX(fechaped) FROM Pedidos pe);
```

	numped	fechaped	codigpro	ivaped	fentrpel
1	5	2010-11-13	P004	21	2010-11-13

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en la proyección:
 - Ejemplo: devolver el número de pedido y proveedor de todos los pedidos, junto con una columna añadida que indique el precio del artículo más caro del pedido.

```
SELECT pe.numped, pe.codigpro,  
       (SELECT MAX(l.preunlin) FROM lineas l WHERE pe.numped = l.numped) as articuloMasCaro  
FROM Pedidos pe;
```

	numped	codigpro	articuloMasCaro
1	1	P001	295
2	2	P002	300
3	3	P003	110
4	4	P001	180
5	5	P004	NULL

Consultas en SQL: Subconsultas

- Se pueden usar subconsultas en el FROM:
 - Ejemplo: devolver el número de pedido y proveedor de todos los pedidos, junto con una columna añadida que indique el precio del artículo más caro del pedido.

```
SELECT pe.numped, pe.codigpro, lm.maximo
FROM Pedidos pe JOIN
    (SELECT l.numped, MAX(l.preunlin) as maximo FROM lineas l
     GROUP BY l.numped) lm
ON pe.numped = lm.numped;
```

	numped	codigpro	maximo
1	1	P001	295
2	2	P002	300
3	3	P003	110
4	4	P001	180

Updates con Subconsultas

- Las subconsultas pueden ser muy útiles para realizar UPDATES “complejos”.
 - Ejemplo: actualizar a día de hoy la fecha de envío de los pedidos del proveedor con CIF ‘A39184215’.

```
UPDATE pedidos  
SET fechaped = date('now')  
WHERE ...
```

¡El CIF de los proveedores está en una tabla diferente a la actualizada!

Updates con Subconsultas

- Las subconsultas pueden ser muy útiles para realizar UPDATES “complejos”.
 - Ejemplo: actualizar a día de hoy la fecha de envío de los pedidos del proveedor con CIF ‘A39184215’.

```
UPDATE pedidos
SET fechaped = date('now')
WHERE codigpro =
    (SELECT pr.codigpro
     FROM Proveedores pr
     WHERE pr.cifpro = 'A39184215')
```

Deletes con Subconsultas

- Las subconsultas pueden ser muy útiles para realizar DELETES “complejos”.
 - Ejemplo: eliminar las líneas de los pedidos realizados entre abril y mayo de 2010

```
DELETE FROM Lineas
WHERE numped IN
    (SELECT pe.numped
     FROM Pedidos pe
     WHERE pe.fechaped BETWEEN '2010-04-01' AND '2010-05-31');
```

AHORA TOCA HACER EL EJERCICIO 9