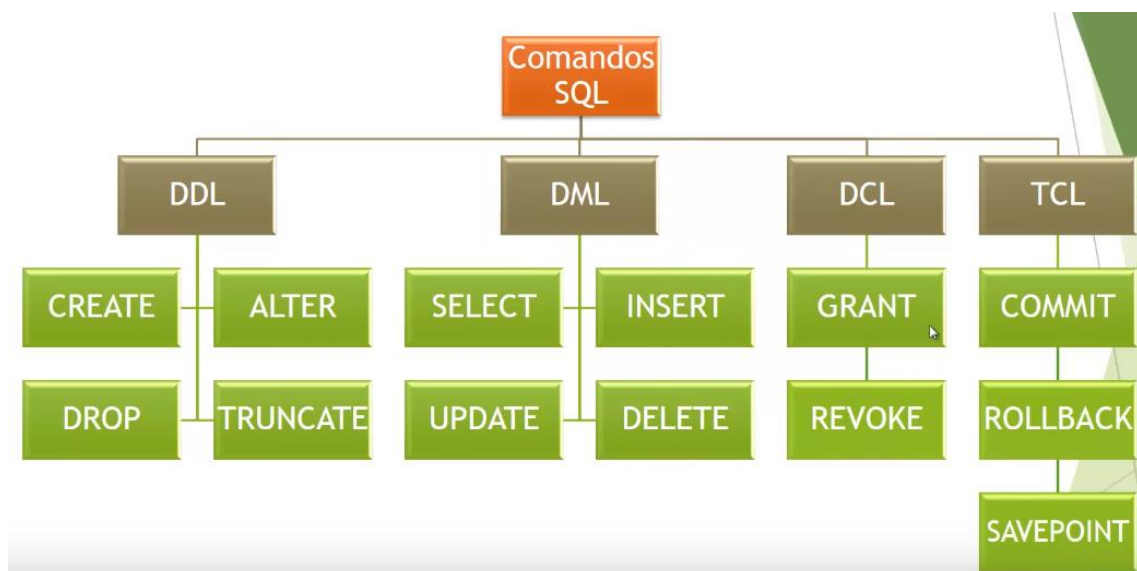


- ▶ Consultas de acción:
 - ▶ Actualización
 - ▶ Creación de tabla
 - ▶ Eliminación
 - ▶ Datos anexados
- ▶ Comandos DML y DDL
 - ▶ Create
 - ▶ Update
 - ▶ Delete
 - ▶ Insert Into
 - ▶ Select into



Consultas de acción

Comandos para ACTUALIZAR:

- Actualizar los CAMPOS de tipo precio, incrementando el precio.

```
UPDATE PRODUCTOS SET PRECIO = PRECIO+10 WHERE SECCIÓN='DEPORTES'
```

- Actualizar nombre de REGISTRO.

```
UPDATE PRODUCTOS SET SECCIÓN = 'DEPORTIVOS' WHERE SECCIÓN = 'DEPORTES'
```

Comandos para CREAR:

- Crear BASE DE DATOS: CREATE DATABASE *nombre*;

- Crear TABLA:

```
create table datosusuarios (nombre varchar(30), clave varchar(10));
```

1. Datosusuarios: Nombre tabla.
2. Nombre: Nombre del primer campo.
3. Varchar: Especificamos el tipo de datos que se guarda (tipo texto de 30 o 10 caracteres en este caso).
4. Clave: Nombre del segundo campo.

Comandos para ELIMINAR:

- **Borrar TABLA:**

```
DROP TABLE DATOSUSUARIOS;
```

- **Borrar BASE DE DATOS:**

```
DROP DATABASE USUARIOS;
```

- **Borrar REGISTRO de una tabla:**

```
DELETE FROM CLIENTES WHERE POBLACIÓN = 'MADRID'
```

- **Borrar REGISTRO con OPERADORES LÓGICOS:**

```
DELETE FROM PRODUCTOS WHERE SECCIÓN = 'DEPORTIVO' AND PRECIO BETWEEN 50 AND 100
```

Comandos para AGREGAR / INSERTAR:

- **Agregar CAMPO a una tabla:**

```
ALTER TABLE datospersonales ADD COLUMN EDAD INT (2);
```

- **Agregar/Insertar DATOS dentro de cada campo:**

```
INSERT INTO datospersonales (NIF, NOMBRE, APELLIDO, EDAD) VALUES ("51729429Y", "Maria", "Gómez", 27);
```

Comandos para ALTERAR:

- **Agregar CAMPO en una tabla:**

```
ALTER TABLE CLIENTES_MADRID ADD COLUMN FECHA_BAJA DATE
```

```
ALTER TABLE PRUEBA ADD COLUMN REGION VARCHAR(20)
```

- **Borrar CAMPO de una tabla:**

```
ALTER TABLE datospersonales DROP EDAD;
```

```
ALTER TABLE PRUEBA DROP COLUMN POBLACION
```

- **Alterar tipo de dato en REGISTRO:**

```
ALTER TABLE clientes_madrid ALTER COLUMN FECHA DATE
```

SQL (Structured Query Language)

- Establecer valor por defecto (en vez de NULL)

```
ALTER TABLE PRUEBA ALTER COLUMN LUGAR_NACIMIENTO SET DEFAULT 'DESCONOCIDO'
```

<input type="checkbox"/>	4	LUGAR_NACIMIENTO	varchar(20)	latin1_swedish_ci	Sí
					DESCONOCIDO

- o Y para borrarlo:

```
ALTER TABLE PRUEBA ALTER COLUMN LUGAR_NACIMIENTO DROP DEFAULT
```

- Indicar CLAVE PRIMARIA al crear una tabla:

```
CREATE TABLE EJEMPLO (DNI TEXT, NOMBRE TEXT, APELLIDO TEXT, EDAD NUMBER, PRIMARY KEY (DNI))
```

- o Indicar clave primaria con la tabla ya creada:

```
ALTER TABLE EJEMPLO ADD PRIMARY KEY (DNI)
```

Comandos de Acces (SELECT INTO):

- Crear tabla COPIANDO datos de otra tabla (solo en Acces)

```
SELECT * INTO CLIENTES_MADRID FROM CLIENTES WHERE POBLACIÓN = 'MADRID'
```

CÓDIGOCLIE	EMPRESA	DIRECCIÓN	POBLACIÓN	TELÉFONO	RESPONSAB	HISTORIAL
CT01	BELTRÁN E HIJOS	LAS FUENTES 7	MADRID	914456435	ANGEL MARTÍN	
CT04	EXPORTASA	VALLECAS 34	MADRID	913452378	ELVIRA GÓMEZ	
CT07	LA CASA DEL JUAN AMÉRICA 45		MADRID	912649987	ELÍAS PÉREZ	
CT10	FERRETERÍA EL PASO DE ALCALA		MADRID	914354866	MANUEL MENÉNDEZ	
CT14	DEPORTES GARCÍA GUZMÁN EL BLANCO		MADRID	913299475	ANA JIMÉNEZ	
CT19	CONFECCIONES FUENCARRAL 7		MADRID	913859234	JUAN GARCÍA	
CT23	EL PALACIO DE ORTEGA Y GASPAR		MADRID	927785235	LAURA CARRASCO	
CT26	FERRETERÍA LA ORENSE 7		MADRID	918459346	JOSÉ GARCÍA	
CT30	BAZAR EL ARGENTINO ATOCHA 55		MADRID	912495973	ADRIÁN ÁLVAREZ	
CT34	BAZAR LA FAROLA CASTILLA Y LEÓN		MADRID	915483627	ANGEL SANTALUCIA	
CT36	JUGUETES EDUCACIÓN ORENSE 89		MADRID	916872354	PEDRO IGLESIA	
CT39	FERRETERÍA LUIS VALLECAS 45		MADRID	913532785	LUIS GARCÍA	

- Lo mismo, pero en MySQL (CREATE TABLE):

```
CREATE TABLE CLIENTES_MADRID SELECT * FROM CLIENTES WHERE POBLACIÓN = 'MADRID'
```

Comandos para VER / SELECT:

- Ver información dentro ALGUNOS campos:

```
SELECT NOMBRE, APELLIDO FROM datospersonales
```

SQL (Structured Query Language)

- Ver información de **TODOS** los campos:

```
SELECT * FROM datospersonales
```

- Ver información de dos **REGISTROS (OR, AND)** dentro de una tabla:

```
SELECT * from productos WHERE SECCIÓN = "CERAMICA" OR SECCIÓN = "DEPORTES"
```

```
SELECT * FROM PRODUCTOS WHERE SECCIÓN = "DEPORTES" AND PAÍSDEORIGEN = "USA"
```

- Ver registros que estén comprendidos **ENTRE DOS** fechas:

```
SELECT * FROM PRODUCTOS WHERE FECHA BETWEEN '2000-03-01' AND '2000-04-30'
```

```
SELECT * FROM PRODUCTOS WHERE FECHA >='2000-03-01' AND FECHA <='2000-04-30'
```

- Ver registros de dos secciones, **ORDENADOS**:

- o Orden ascendente:

```
SELECT * FROM PRODUCTOS WHERE SECCIÓN = "DEPORTES" OR SECCIÓN = "CERAMICA" ORDER BY FECHA
```

- o Orden descendente:

```
SELECT * FROM PRODUCTOS WHERE SECCIÓN = "DEPORTES" OR SECCIÓN = "CERAMICA" ORDER BY FECHA desc
```

- o Ordenar por dos registros:

```
SELECT * FROM PRODUCTOS WHERE SECCIÓN = 'DEPORTES' OR SECCIÓN = 'CERÁMICA' ORDER BY SECCIÓN, PRECIO
```

Comandos Datos Anexados:

- o Nos va a permitir, anexas/unir los datos de una tabla, en otra.

- Anexas/Unir toda la tabla de **CLIENTE_MADRID**, dentro de la tabla **CLIENTES**:

(En Acces y MYSQL) `INSERT INTO CLIENTES SELECT * FROM CLIENTES_MADRID`

- Lo mismo que lo anterior, pero especificando los campos que queremos anexar.

- o Sin la DIRECCIÓN y RESPONSABLE

```
INSERT INTO CLIENTES (CÓDIGOCLIENTE, EMPRESA, POBLACIÓN, TELÉFONO)
SELECT CÓDIGOCLIENTE, EMPRESA, POBLACIÓN, TELÉFONO FROM CLIENTES_MADRID
```

Comandos usando LIKE:

- Encontrar apellidos LIKE Gómez.

```
SELECT * FROM datospersonales WHERE APELLIDO LIKE 'Gómez%'
```

- Si alguien se equivoca y en vez de poner GarCía, ponen GarZia, la forma de descartar esa letra para encontrar los resultados es la siguiente:

```
SELECT * FROM `datospersonales` WHERE APELLIDO LIKE 'Gar_ía'
```

NIF	NOMBRE	APELLIDO	EDAD
48171160Y	Mario	García	22
48171160Y	Mario	Garzía	22

Consultas por agrupación

Función	Descripción
AVG	Calcula el promedio de un campo
COUNT	Cuenta los registros de un campo
SUM	Suma los valores de un campo
MAX	Devuelve el máximo de un campo
MIN	Devuelve el mínimo de un campo

Función SUM (Suma)

1. (SUM) Agrupar campos y sumar el precio de las agrupaciones:

```
SELECT SECCIÓN, SUM(PRECIO) FROM PRODUCTOS GROUP BY SECCIÓN
```

SECCION	SUM(PRECIO)
CERÁMICA	440
CONFECCIÓN	2042
DEPORTES	1302
FERRETERÍA	93
JUGUETERÍA	1010.505
OFICINA	39

1.1. (SUM) Ordenar lo anterior por precio.

- Si ejecutamos lo siguiente dará error:

```
SELECT SECCIÓN, SUM(PRECIO) FROM PRODUCTOS GROUP BY SECCIÓN ORDER BY PRECIO
```

Debido a que en la tabla resultante no tenemos ningún campo que se llame precio. Y lo que hace la cláusula ORDER BY es buscar el registro en la consulta, no en la tabla.

- Consulta:

SECCION	SUM(PRECIO)
CERÁMICA	440
CONFECCIÓN	2042
DEPORTES	1302
FERRETERÍA	93
JUGUETERÍA	1010.505
OFICINA	39

SQL (Structured Query Language)

Se puede solucionar utilizando un **ALIAS**, que nos permite cambiar el nombre a un campo. Le cambiamos el nombre a **SUM(PRECIO)** por **SUMA**, y aplicamos el orden a **SUMA**.

```
SELECT SECCIÓN, SUM(PRECIO) AS SUMA FROM PRODUCTOS GROUP BY SECCIÓN ORDER BY SUMA
```

SECCIÓN	SUMA
OFICINA	39
FERRETERÍA	93
CERÁMICA	440
JUGUETERÍA	1010.505
DEPORTES	1302
CONFECCIÓN	2042

1.2. **(SUM)** Además de lo anterior, queremos que nos aparezca el primer elemento de cada campo (de la tabla).

```
SELECT SECCIÓN, NOMBREARTÍCULO, SUM(PRECIO) AS SUMA FROM PRODUCTOS GROUP BY SECCIÓN ORDER BY SUMA
```

SECCIÓN	NOMBREARTÍCULO	SUMA
OFICINA	PIE DE LÁMPARA	39
FERRETERÍA	DESTORNILLADOR	93
CERÁMICA	TUBOS	440
JUGUETERÍA	COCHE TELEDIRIGIDO	1010.505
DEPORTES	RAQUETA TENIS	1302
CONFECCIÓN	TRAJE CABALLERO	2042

Función AVG (Promedio)

1. **(AVG)** Sacar la media del precio de todos los registros que tengan como campo **SECCIÓN** el nombre registro de: **DEPORTES** o **CONFECCIÓN**:

- **HAVING** sustituye a **WHERE** en consultas de agrupación.

```
SELECT SECCIÓN, AVG(PRECIO) AS MEDIA_ARTICULOS FROM PRODUCTOS GROUP BY SECCIÓN HAVING SECCIÓN='DEPORTES' OR SECCIÓN='CONFECCIÓN'
```

SECCIÓN	MEDIA_ARTICULOS
CONFECCIÓN	226.88888888888889
DEPORTES	130.2

Función COUNT (Contar)

1. (COUNT) Cuantos clientes tengo por cada provincia:

- La función COUNT solamente cuenta los registros de los campos que están rellenos.

```
SELECT POBLACIÓN, COUNT(CÓDIGOCLIENTE) AS N_CLIENTES FROM CLIENTES GROUP BY POBLACIÓN
```

POBLACIÓN	N_CLIENTES
A CORUÑA	1
ÁVILA	1
BARCELONA	8
BILBAO	1
GIJÓN	2
LEÓN	1
LUGO	1
MADRID	12
MÁLAGA	1
OVIEDO	1
SALAMANCA	2
SANTANDER	1
SEVILLA	2
TERUEL	1
ZAMORA	1
ZARAGOZA	2

Función MAX (Máximo)

1. (MAX) Ver el precio más alto entre todos los productos cuya SECCIÓN sea CONFECCIÓN:

```
SELECT SECCIÓN, MAX(PRECIO) AS PRECIO_MAS_ALTO FROM PRODUCTOS WHERE SECCIÓN = 'CONFECCIÓN' GROUP BY SECCIÓN
```

SECCIÓN	PRECIO_MAS_ALTO
CONFECCIÓN	67,13

Consultas multitabla

► Consultas Multitabla / Consultas de Unión

► Unión Externa:

- Union
- Union All
- Except
- Intersect
- Minus

No todos los gestores de BBDD aceptan este tipo de instrucciones.
MySQL y Acces no los aceptan

► Unión interna

- Inner join
- Left Join
- Right Join

- Deben de tener campos compatibles y el mismo número de campos.

Tabla 1

CÓDIGO ARTÍCULO	SECCIÓN	NOMBRE ARTÍCULO	PRECIO	FECHA	IMPORTADO	PAÍS DE ORIGEN
AR01	FERRETERÍA	DESTORNILLADOR	6,63 €	22/10/2000		ESPAÑA
AR02	CONFECCIÓN	TRAJE CABALLERO	284,58 €	11/03/2002	✓	ITALIA
AR03	JUGUETERÍA	COCHE TELEDIRIGIDO	159,45 €	26/05/2002	✓	MARRUECOS
AR04	DEPORTES	RAQUETA TENIS	92,47 €	20/02/2000	✓	USA
AR06	DEPORTES	MANCUERNAS	60,00 €	13/09/2000	✓	USA
AR07	CONFECCIÓN	SERRUCHO	30,20 €	23/03/2001	✓	FRANCIA
AR08	JUGUETERÍA	CORREPASILLOS	103,34 €	11/04/2000	✓	JAPÓN
AR09	CONFECCIÓN	PANTALÓN SEÑORA	174,23 €	10/01/2000	✓	MARRUECOS
AR10	JUGUETERÍA	CONSOLA VIDEO	442,54 €	24/09/2002	✓	USA
AR11	CERÁMICA	TUBOS	168,43 €	04/02/2000	✓	CHINA
AR12	FERRETERÍA	LLAVE INGLESA	24,40 €	23/05/2001	✓	USA
AR13	CONFECCIÓN	CAMISA CABALLERO	67,13 €	11/08/2002		ESPAÑA
AR14	JUGUETERÍA	TREN ELÉCTRICO	1.505,38 €	03/07/2001	✓	JAPÓN
AR15	CERÁMICA	PLATO DECORATIVO	54,09 €	07/06/2000	✓	CHINA
AR16	FERRETERÍA	ALICATES	6,74 €	17/04/2000	✓	ITALIA

Tabla 2

CÓDIGO ART	SECCIÓN	NOMBRE ARTÍCULO	PRECIO	FECHA	IMPORTADO	PAÍS DE ORIGEN	FOTO
ARS0	ALTA COSTURA	TRAJE CABALLERO	1.284,58 €	11/03/2002		-1 ITALIA	
ARS1	DEPORTES DE RIESGO	RAQUETA TENIS	1.093,47 €	20/03/2000		-1 USA	
ARS2	DEPORTES DE RIESGO	MANCUERNAS	1.060,00 €	13/09/2000		-1 USA	
ARS3	ALTA COSTURA	SERRUCHO	1.030,20 €	23/03/2001		-1 FRANCIA	
ARS4	ALTA COSTURA	PANTALÓN SEÑORA	1.174,23 €	10/01/2000		-1 MARRUECOS	
ARS5	ALTA COSTURA	CAMISA CABALLERO	1.067,13 €	11/08/2002		0 ESPAÑA	
ARS6	DEPORTES DE RIESGO	PISTOLA OLÍMPICA	1.046,73 €	02/02/2001		-1 SUECIA	
ARS7	ALTA COSTURA	BLUSA SRA.	1.101,06 €	18/03/2000		-1 CHINA	
ARS8	ALTA COSTURA	CAZADORA PIEL	1.522,69 €	10/07/2001		-1 ITALIA	
ARS9	DEPORTES DE RIESGO	BALÓN RUGBY	1.111,64 €	11/11/2000		-1 USA	
AR60	DEPORTES DE RIESGO	BALÓN BALONCESTO	1.075,27 €	25/06/2001		-1 JAPÓN	
AR61	ALTA COSTURA	ABRIGO CABALLERO	1.500,00 €	05/04/2002		-1 ITALIA	
AR62	DEPORTES DE RIESGO	BALÓN FÚTBOL	1.043,91 €	04/07/2002		0 ESPAÑA	
AR63	ALTA COSTURA	ABRIGO SRA	1.360,07 €	03/05/2001		-1 MARRUECOS	
AR64	DEPORTES DE RIESGO	CRONÓMETRO	1.439,18 €	03/01/2002		-1 USA	

- El campo PRECIO, por ejemplo, no puede ser de tipo VARCHAR en una tabla, y en la otra de tipo numérico.

Consultas multitabla (UNIÓN EXTERNA)

- **UNION:** En caso que hay registros repetidos en tabla 1 y tabla 2, esos registros solo los muestra una vez.
- **UNION ALL:** En caso de haber registros repetidos en tablas, los muestra tantas veces como estén contemplados en las tablas.

1. (UNION) Buscar registros dentro de dos campos iguales (SECCIÓN) en dos tablas distintas:

```
SELECT * FROM productos WHERE SECCIÓN = 'DEPORTES' UNION SELECT * FROM productosnuevos WHERE SECCIÓN = 'DEPORTES DE RIESGO'
```

CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	FECHA	IMPORTADO	PAÍSDEORIGEN	FOTO
AR32	DEPORTES	CRONÓMETRO	439,18	2002-01-03	VERDADERO	USA	NULL
AR38	DEPORTES	CAÑA DE PESCA	270,00	2000-02-14	VERDADERO	USA	NULL
AR40	DEPORTES	BOTA ALPINISMO	144,00	2002-05-05	FALSO	ESPAÑA	NULL
AR41	DEPORTES	PALAS DE PING PONG	21,60	2002-02-02	FALSO	ESPAÑA	NULL
AR51	DEPORTES DE RIESGO	RAQUETA TENIS	1.093,47	20/03/2000	VERDADERO	USA	NULL
AR52	DEPORTES DE RIESGO	MANCUERNAS	1.060,00	13/09/2000	VERDADERO	USA	NULL
AR56	DEPORTES DE RIESGO	PISTOLA OLÍMPICA	1.046,73	02/02/2001	VERDADERO	SUECIA	NULL
AR59	DEPORTES DE RIESGO	BALÓN RUGBY	1.111,64	11/11/2000	VERDADERO	USA	NULL

1.1. (UNION) Buscar registros de una tabla cuyo precio sea mayor a 500, y todos los productos de otra tabla cuya sección sea ALTA COSTURA.

```
SELECT * FROM PRODUCTOS WHERE PRECIO > 500 UNION SELECT * FROM productosnuevos WHERE SECCIÓN = 'ALTA COSTURA'
```

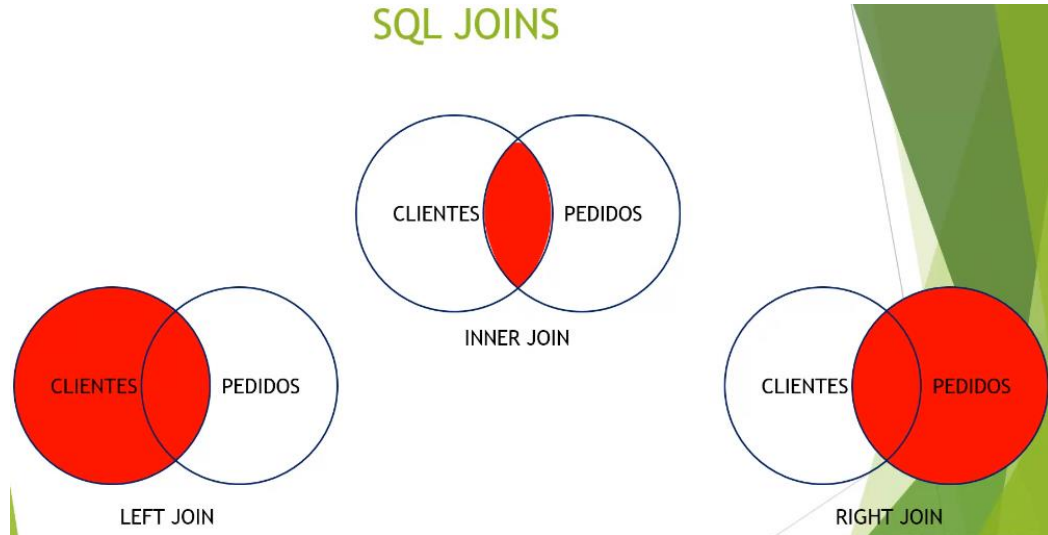
CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	FECHA	IMPORTADO	PAÍSDEORIGEN	FOTO
AR23	CONFECCIÓN	CAZADORA PIEL	522,69	2001-07-10	VERDADERO	ITALIA	NULL
AR50	ALTA COSTURA	TRAJE CABALLERO	1.284,58	11/03/2002	VERDADERO	ITALIA	NULL
AR53	ALTA COSTURA	SERRUCHO	1.030,20	23/03/2001	VERDADERO	FRANCIA	NULL
AR54	ALTA COSTURA	PANTALÓN SEÑORA	1.174,23	10/01/2000	VERDADERO	MARRUECOS	NULL
AR55	ALTA COSTURA	CAMISA CABALLERO	1.067,13	11/08/2002	FALSO	ESPAÑA	NULL
AR57	ALTA COSTURA	BLUSA SRA.	1.101,06	18/03/2000	VERDADERO	CHINA	NULL

1. (UNION ALL) Buscar todos los registros de dos tablas teniendo en cuenta los registros repetidos (para que se muestren).

```
SELECT * FROM productos WHERE SECCIÓN = 'DEPORTES' UNION ALL SELECT * FROM productosnuevos
```

Consultas multitabla (UNIÓN INTERNA)

SQL JOINS: Nos devuelve registros que se contemplan en las tablas.

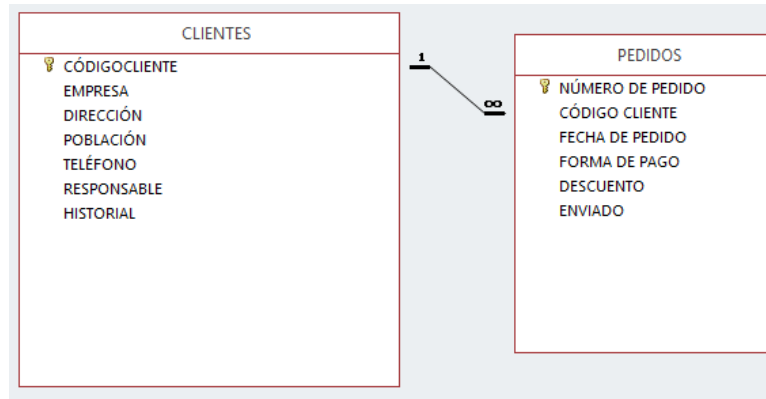


- **INNER JOIN:** Nos devuelve únicamente los registros que están contemplados en las dos tablas.
- **LEFT JOIN:** Nos devuelve los registros contemplados en las dos tablas, más los que están 'en la tabla de la izquierda'.
- **RIGHT JOIN:** Nos devuelve los registros contemplados en las dos tablas, más los que están 'en la tabla de la derecha'.

SQL (Structured Query Language)

1. (INNER JOIN) Buscar clientes de MADRID que hayan hecho pedidos.

- Hay que tener en cuenta cual es el campo que relaciona cada tabla:
(En este caso CÓDIGO CLIENTE).



También hay que tener en cuenta que en una tabla CÓDIGO CLIENTE se escribe junto, y en la otra no, por eso hay que escribir el que va separado entre [].

```
SELECT * FROM CLIENTES INNER JOIN PEDIDOS ON CLIENTES.CÓDIGOCLIENTE=PEDIDOS.[CÓDIGO CLIENTE] WHERE POBLACIÓN='MADRID'
```

- Traducción del código:
Seleccionar todos desde **Clientes** -> **INNER JOIN** <- Pedidos donde el campo **CÓDIGOCLIENTE** (de clientes) sea igual que **CÓDIGO CLIENTE**, donde la población sea Madrid.

1. (LEFT JOIN) Buscar clientes de MADRID que hayan hechos pedidos o NO (teniendo en cuenta que aparecerán todos los clientes, hayan hecho pedidos o no).

```
SELECT * FROM CLIENTES LEFT JOIN PEDIDOS ON CLIENTES.CÓDIGOCLIENTE=PEDIDOS.[CÓDIGO CLIENTE] WHERE POBLACIÓN = 'MADRID'
```

CÓDIGOCLIE	EMPRESA	DIRECCIÓN	POBLACIÓN	TELÉFONO	RESPONSABLE	HISTORIAL	NÚMERO DE PEDID	CÓDIGO CLIENTE	FECHA DE PEDIDC	FORMA DE PAGC	DESCUENTO	ENVIAD
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		1	CT01	11/03/2000	CONTADO	2,00%	<input checked="" type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		8	CT01	15/04/2000	TARJETA	2,00%	<input checked="" type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		72	CT01	18/08/2002	CONTADO	5,00%	<input checked="" type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		73	CT01	02/08/2001	CONTADO	7,00%	<input type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		74	CT01	17/09/2002	APLAZADO	8,00%	<input type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		75	CT01	30/09/2002	TARJETA	12,00%	<input type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		76	CT01	19/10/2002	CONTADO	4,00%	<input checked="" type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		77	CT01	28/10/2000	CONTADO	5,00%	<input type="checkbox"/>
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	(914) 45- 64- 35	ANGEL MARTÍNEZ		98	CT01	27/12/2001	CONTADO	8,00%	<input checked="" type="checkbox"/>
CT04	EXPORTASA	VALLECAS 34	MADRID	(913) 45- 23- 78	ELVIRA GÓMEZ		11	CT04	01/05/2001	CONTADO	8,00%	<input checked="" type="checkbox"/>
CT04	EXPORTASA	VALLECAS 34	MADRID	(913) 45- 23- 78	ELVIRA GÓMEZ		40	CT04	12/07/2002	CONTADO	12,00%	<input type="checkbox"/>
CT04	EXPORTASA	VALLECAS 34	MADRID	(913) 45- 23- 78	ELVIRA GÓMEZ		85	CT04	23/12/2002	TARJETA	4,00%	<input type="checkbox"/>
CT07	LA CASA DEL JUGUETE	AMÉRICA 45	MADRID	(912) 64- 99- 87	ELÍAS PÉREZ		22	CT07	31/05/2000	TARJETA	5,00%	<input checked="" type="checkbox"/>
CT10	FERRETERÍA EL CLAVO	PASEO DE ÁLAMOS 78	MADRID	(914) 35- 48- 66	MANUEL MENÉNDEZ		19	CT10	22/05/2002	CONTADO	7,00%	<input checked="" type="checkbox"/>
CT14	DEPORTES GARCÍA	GUZMÁN EL BUENO 45	MADRID	(913) 29- 94- 75	ANA JIMÉNEZ		32	CT14	20/06/2001	APLAZADO	6,00%	<input type="checkbox"/>
CT19	CONFECCIONES GALÁN	FUENCARRAL 78	MADRID	(913) 85- 92- 34	JUAN GARCÍA							<input checked="" type="checkbox"/>
CT23	EL PALACIO DE LA MODA	ORTEGA Y GASSET 129	MADRID	(927) 78- 52- 35	LAURA CARRASCO		3	CT23	18/03/2000	APLAZADO	6,00%	<input type="checkbox"/>

- Si nos fijamos, CT19, no ha hecho ningún pedido, pero aun y así, sale en los resultados

1. (RIGHT JOIN) Buscar clientes de MADRID que hayan hechos pedidos o NO (teniendo en cuenta la tabla de pedidos, cosa que no se puede realizar ya que no se pueden mostrar los pedidos de las personas que no los han hecho).

Subconsultas

Consultas dentro de otra consulta

- ▶ Subconsulta escalonada
- ▶ Subconsulta de lista
- ▶ Subconsulta correlacionada

Subconsulta escalonada

La consulta interna devuelve **una única columna con un único registro** y eso se utiliza como único criterio o comparación en la consulta padre:

1. Ver **NOMBREARTÍCULO, SECCIÓN** de los productos que su precio sea mayor a: *El precio de media que tienen todos los productos de la tabla.*

```
SELECT NOMBREARTÍCULO, PRECIO FROM productos WHERE PRECIO > (SELECT AVG(PRECIO) FROM PRODUCTOS)
```

NOMBREARTÍCULO	PRECIO
TRAJE CABALLERO	284,58
COCHE TELEDIRIGIDO	159,45
PANTALÓN SEÑORA	174,23
CONSOLA VIDEO	442,54
TUBOS	168,43
CAZADORA PIEL	522,69
FUERTE DE SOLDADOS	143,70
ABRIGO CABALLERO	500.000,00
ABRIGO SRA	360,07
CRONOMETRO	439,18
CAÑA DE PESCA	270,00
JARRA CHINA	127,77
BOTA ALPINISMO	144,00

AVG(PRECIO)
123.162625

Subconsulta de lista

La consulta interna devuelve **una lista de registros** (suele utilizar IN, ANY, ALL):

1. **(ALL)** Mostrar los artículos de la tabla de PRODUCTOS cuyo precio sea superior a **TODO**s los artículos de CERÁMICA.

```
SELECT * FROM PRODUCTOS WHERE PRECIO > ALL (SELECT PRECIO FROM PRODUCTOS WHERE SECCIÓN = 'CERÁMICA')
```

CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	FECHA
AR02	CONFECCIÓN	TRAJE CABALLERO	284,58 €	11/03/2002
AR09	CONFECCIÓN	PANTALÓN SEÑORA	174,23 €	10/01/2000
AR10	JUGUETERÍA	CONSOLA VIDEO	442,54 €	24/09/2002
AR14	JUGUETERÍA	TREN ELÉCTRICO	1.505,38 €	03/07/2001
AR23	CONFECCIÓN	CAZADORA PIEL	522,69 €	10/07/2001
AR27	CONFECCIÓN	ABRIGO CABALLERO	500.000,00 €	05/04/2002
AR29	CONFECCIÓN	ABRIGO SRA	360,07 €	03/05/2001
AR32	DEPORTES	CRONÓMETRO	439,18 €	03/01/2002
AR38	DEPORTES	CAÑA DE PESCA	270,00 €	14/02/2000

PRECIO

168,43

54,09

43,27

19,75

29,04

127,77

2. **(ANY)** Mostrar los artículos de la tabla de PRODUCTOS cuyo precio sea superior **CUALQUIERA** los artículos de CERÁMICA.

- (Es decir, los productos que tengan más precio en comparación al que tiene el precio menor entre los productos de CERÁMICA)

```
SELECT * FROM PRODUCTOS WHERE PRECIO > ANY (SELECT PRECIO FROM PRODUCTOS WHERE SECCIÓN = 'CERÁMICA')
```

CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	F
AR02	CONFECCIÓN	TRAJE CABALLERO	284,58 €	1
AR03	JUGUETERÍA	COCHE TELEDIRIGIDO	159,45 €	2
AR04	DEPORTES	RAQUETA TENIS	93,47 €	2
AR06	DEPORTES	MANCUERNAS	60,00 €	1
AR07	CONFECCIÓN	SERRUCHO	30,20 €	2
AR08	JUGUETERÍA	CORREPASILLOS	103,34 €	1
AR09	CONFECCIÓN	PANTALÓN SEÑORA	174,23 €	1
AR10	JUGUETERÍA	CONSOLA VIDEO	442,54 €	2
AR11	CERÁMICA	TUBOS	168,43 €	0
AR12	FERRETERÍA	LLAVE INGLESA	24,40 €	2
AR13	CONFECCIÓN	CAMISA CABALLERO	67,13 €	1
AR14	JUGUETERÍA	TREN ELÉCTRICO	1.505,38 €	0
AR15	CERÁMICA	PLATO DECORATIVO	54,09 €	0
AR17	JUGUETERÍA	MUÑECA ANDADORA	105,06 €	0
AR18	DEPORTES	PISTOLA OLÍMPICA	46,73 €	0
AR19	CONFECCIÓN	BLUSA SRA.	101,06 €	1

PRECIO

168,43

54,09

43,27

19,75

29,04

127,77

SQL (Structured Query Language)

3. **(IN)** Mostrar **NOMBREARTÍCULO + PRECIO** de la tabla **PEDIDOS**, de los productos que tengan más de 20 unidades en la tabla **PRODUCTOS-PEDIDOS**.

```
SELECT NOMBREARTÍCULO, PRECIO FROM PRODUCTOS WHERE CÓDIGOARTÍCULO IN (SELECT [CÓDIGO ARTÍCULO] FROM [PRODUCTOS- PEDIDOS] WHERE UNIDADES > 20)
```

NOMBREARTÍCULO	PRECIO
LLAVE INGLESA	24,40 €
CENICERO	19,75 €
MARTILLO	11,40 €
CAZADORA PIEL	522,69 €
BALÓN RUGBY	111,64 €
BALÓN FÚTBOL	43,91 €
ABRIGO SRA	360,07 €
PISTOLA CON SONIDOS	57,25 €
LIMA GRANDE	22,07 €
JUEGO DE BROCAS	15,10 €

CÓDIGO ARTÍCULO
AR12
AR35
AR28
AR21
AR23
AR24
AR29
AR31
AR22
AR36

- Consulta anterior con consulta multitabla **INNER JOIN**:

```
SELECT NOMBREARTÍCULO, PRECIO FROM PRODUCTOS INNER JOIN [PRODUCTOS- PEDIDOS] ON PRODUCTOS.CÓDIGOARTÍCULO = [PRODUCTOS- PEDIDOS].[CÓDIGO ARTÍCULO] WHERE UNIDADES > 20
```

- 3.1. **(NOT IN)** Mostrar **CLIENTES** que **NO** han hecho pedidos con tarjeta o que **NO** han hecho pedidos.

```
SELECT EMPRESA, POBLACIÓN FROM CLIENTES WHERE CÓDIGOCIENTE NOT IN (SELECT [CÓDIGO CLIENTE] FROM PEDIDOS WHERE [FORMA DE PAGO] = "TARJETA" AND [NÚMERO DE PEDIDO] > 0)
```

EMPRESA	POBLACIÓN
LA MODERNA	OVIEDO
EL ESPAÑOLITO	BARCELONA
CONFECCIONES AMPARO	GIJÓN
JUGUETERÍA SUÁREZ	BARCELONA
ALMACÉN POPULAR	BILBAO
FERRETERÍA EL CLAVO	MADRID
JUGUETES MARTÍNEZ	BARCELONA
FERNÁNDEZ SL	SANTANDER
CONFECCIONES ARTÍMEZ	A CORUÑA
DEPORTES GARCÍA	MADRID
EXCLUSIVAS FERNÁNDEZ	BARCELONA
DEPORTES MORÁN	LUGO
BAZAR FRANCISCO	ZAMORA
JUGUETES LA SONRISA	LEÓN
CONFECCIONES GALÁN	MADRID
LÍNEA JOVEN	SEVILLA
BAZAR EL BARAT	BARCELONA
EL PALACIO DE LA MODA	MADRID
DEPORTES EL MADRILEÑO	ZARAGOZA
JUGUETES EL BARATO	BARCELONA
CONFECCIONES HERMINIA	GIJÓN
LA TIENDA ELEGANTE	ZARAGOZA
DEPORTES NAUTICOS GARCÍA	ÁVILA
CONFECCIONES RUIZ	BARCELONA
BAZAR LA FARAONA	MADRID
FERRETERÍA EL MARTILLO	SALAMANCA
JUGUETES EDUCATIVOS SANZ	MADRID
ALMACENES FERNANDEZ	TERUEL
CONFECCIONES MÓNICA	BARCELONA
FERRETERÍA LIMA	MADRID
DEPORTES EL BRASILEÑO	SALAMANCA

CÓDIGO CLIENTE
CT01
CT07
CT30
CT30
CT26
CT24
CT20
CT30
CT01
CT04
CT30
CT30

Predicados

- **DISTINCT:** Poniéndolo en la consulta; delante de un campo que **no** queremos que se repita.
- **DISTINCTROW:** Poniéndolo en la consulta; descarta las filas que sean que estén repetidas.

1. (DISTINCT) Saber los CLIENTES que han hecho pedidos (SIN REPETIR CAMPOS).

- Consulta de INNER JOIN para saber los que han hecho pedidos:

```
SELECT EMPRESA FROM CLIENTES INNER JOIN PEDIDOS ON CLIENTES.CÓDIGOCLIENTE = PEDIDOS.[CÓDIGO CLIENTE]
```

EMPRESA
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
BELTRÁN E HIJOS
LA MODERNA
LA MODERNA
EXPORTASA
EXPORTASA
EXPORTASA
CONFECCIONES AMPARO
CONFECCIONES AMPARO
LA CASA DEL JUGUETE
ALMACÉN POPULAR
ALMACÉN POPULAR
ALMACÉN POPULAR

- Consulta como la anterior, pero añadiendo DISTINCT para evitar repetidos.

```
SELECT DISTINCT EMPRESA FROM CLIENTES INNER JOIN PEDIDOS ON CLIENTES.CÓDIGOCLIENTE = PEDIDOS.[CÓDIGO CLIENTE]
```

EMPRESA
ALMACÉN POPULAR
BAZAR EL ARGENTINO
BAZAR LA FARAONA
BELTRÁN E HIJOS
CONFECCIONES AMPARO
CONFECCIONES ARTÍMEZ
CONFECCIONES HERMINIA
DEPORTES EL MADRILEÑO
DEPORTES GARCÍA
DEPORTES MORÁN
EL PALACIO DE LA MODA
EXPORTASA

2. (DISTINCTROW) Hacer consulta para mostrar registros de tipo FERRETERÍA, sin repetir filas:

- Sin tener en cuenta los repetidos (sin DISTINCTROW):

```
SELECT * FROM productos WHERE SECCIÓN = 'FERRETERIA'
```

CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	FECHA	IMPORTADO	PAISDEORIGEN	FOTO
AR01	FERRETERIA	DESTORNILLADOR	6,63	2000/10/22	FALSO	ESPAÑA	NULL
AR12	FERRETERIA	LLAVE INGLESA	24,40	2001/05/23	VERDADERO	USA	NULL
AR16	FERRETERIA	ALICATES	6,74	2000/04/17	VERDADERO	ITALIA	NULL
AR22	FERRETERIA	MARTILLO	11,40	2001/09/04	FALSO	ESPAÑA	NULL
AR30	FERRETERIA	DESTORNILLADOR	9,06	2002/02/20	VERDADERO	FRANCIA	NULL
AR35	FERRETERIA	LIMA GRANDE	22,07	2002/08/10	FALSO	ESPAÑA	NULL
AR36	FERRETERIA	JUEGO DE BROCAS	15,10	2002/07/04	VERDADERO	TAIWAN	NULL
AR01	FERRETERIA	DESTORNILLADOR	6,63	2000/10/22	FALSO	ESPAÑA	NULL

- Usando DISTINCTROW:

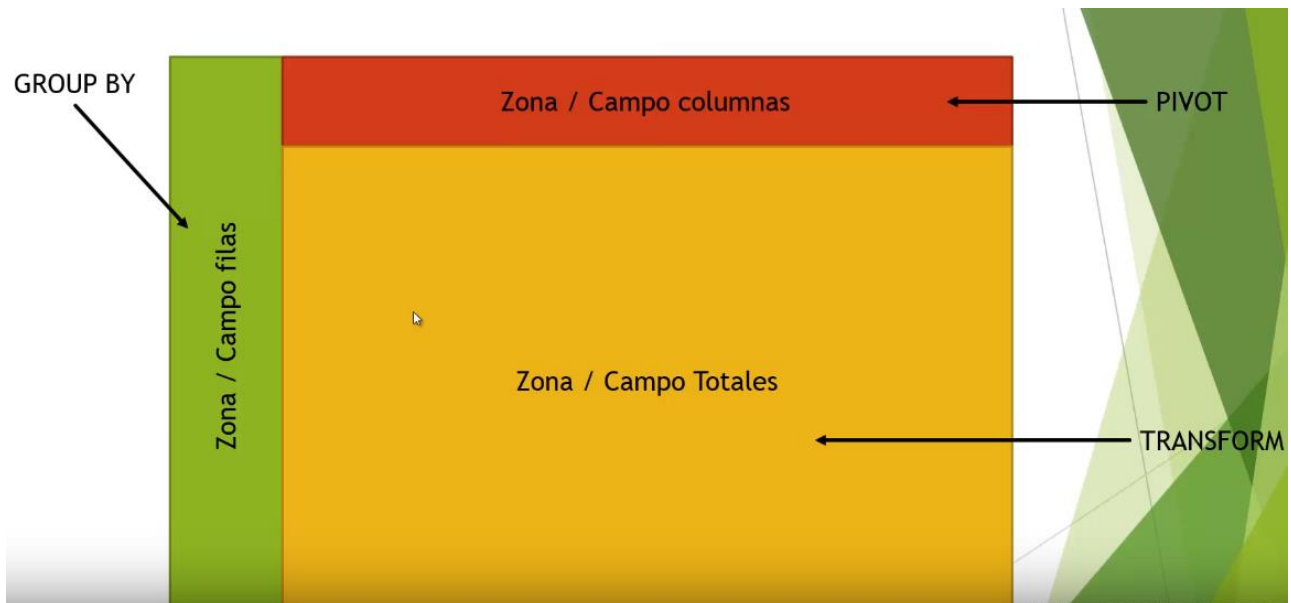
CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	FECHA	IMPORTADO	PAISDEORIGEN	FOTO
AR01	FERRETERIA	DESTORNILLADOR	6,63	2000/10/22	FALSO	ESPAÑA	NULL
AR12	FERRETERIA	LLAVE INGLESA	24,40	2001/05/23	VERDADERO	USA	NULL
AR16	FERRETERIA	ALICATES	6,74	2000/04/17	VERDADERO	ITALIA	NULL
AR22	FERRETERIA	MARTILLO	11,40	2001/09/04	FALSO	ESPAÑA	NULL
AR30	FERRETERIA	DESTORNILLADOR	9,06	2002/02/20	VERDADERO	FRANCIA	NULL
AR35	FERRETERIA	LIMA GRANDE	22,07	2002/08/10	FALSO	ESPAÑA	NULL
AR36	FERRETERIA	JUEGO DE BROCAS	15,10	2002/07/04	VERDADERO	TAIWAN	NULL

2.1. (DISTINCTROW y LEFTJOIN) Borrar todos los CLIENTES que no hayan hecho PEDIDOS.

- Se hace uso de la función LEFT JOIN (para crear una tabla con los CLIENTES que hayan hecho PEDIDOS, y los que no)

```
DELETE DISTINCTROW CLIENTES.*, PEDIDOS.[CODIGO CLIENTE]
FROM CLIENTES LEFT JOIN PEDIDOS ON CLIENTES.CÓDIGOCLIENTE=PEDIDOS.[CÓDIGO CLIENTE]
WHERE PEDIDOS.[CÓDIGO CLIENTE] IS NULL
```

Referencias cruzadas



1. Crear una tabla con referencias cruzadas para saber PRECIOS de artículo.

- En la zona de columnas (**PIVOT**) agrupamos todas las SECCIONES.
- En la zona de filas (**GROUP BY**) agrupamos los NOMBREARTÍCULO.
- En la zona de totales (**TRANSFORM**) agrupamos la suma de los precios de los NOMBREARTÍCULOS (por si acaso hay más de uno, como, por ejemplo, DESTORNILLADOR, que hay 2).

TRANSFORM SUM(PRECIO) AS TOTAL

SELECT NOMBREARTÍCULO FROM PRODUCTOS

GROUP BY NOMBREARTÍCULO

PIVOT SECCIÓN

NOMBREARTÍCULO	CERÁMICA	CONFECCIÓ	DEPORTIVO	FERRETERÍA	JUGUETERÍA	OFICINA
ABRIGO CABALLERO		500.000,00 €				
ABRIGO SRA		360,07 €				
ALICATES				6,74 €		
BALÓN BALONCESTO			75,27 €			
BALÓN FÚTBOL			43,91 €			
BALÓN RUGBY			111,64 €			
BLUSA SRA.		101,06 €				
BOTA ALPINISMO			144,00 €			
CAMISA CABALLERO		67,13 €				
CAÑA DE PESCA			270,00 €			
CAZADORA PIEL		522,69 €				
CENICERO	19,75 €					
CINTURÓN DE PIEL		4,33 €				
COCHE TELEDIRIGIDO					159,45 €	
CONSOLA VIDEO					442,54 €	
CORREPASILLOS					103,34 €	
CRONÓMETRO			439,18 €			
DESTORNILLADOR				15,69 €		
FUERTE DE SOLDADOS					143,70 €	

SQL (Structured Query Language)

1.1. Crear una tabla con referencias cruzadas para saber UNIDADES de cada artículo.

```
TRANSFORM COUNT(NOMBREARTÍCULO) AS N_ARTICULOS
```

```
SELECT NOMBREARTÍCULO FROM PRODUCTOS
```

```
GROUP BY NOMBREARTÍCULO
```

```
PIVOT SECCIÓN
```

```
|
```

- *Vemos que tenemos 2 unidades de DESTORNILLADOR.*

NOMBREARTÍCULO	CERÁMICA	CONFECCIÓN	DEPORTIVO	FERRETERÍA	JUGUETERÍA	OFICINA
BLUSA SRA.		1				
BOTA ALPINISMO			1			
CAMISA CABALLERO		1				
CAÑA DE PESCA			1			
CAZADORA PIEL		1				
CENICERO	1					
CINTURÓN DE PIEL		1				
COCHE TELEDIRIGIDO					1	
CONSOLA VIDEO					1	
CORREPASILLOS					1	
CRONÓMETRO			1			
DESTORNILLADOR				2		
FUERTE DE SOLDADOS					1	
JARRA CHINA	1					
JUEGO DE BROCAS				1		
JUEGO DE TF	1					

SQL (Structured Query Language)

1.2. Contar los pagos que nos han hecho los clientes, y como los han hecho.

- Anteriormente hemos hecho una consulta de tipo INNER JOIN y se ha guardado la tabla con el nombre **PREVIA**.

CLIENTES
CLIENTES_MADRID
PEDIDOS
PRODUCTOS
PRODUCTOS- PEDIDOS
PREVIA

```
SELECT EMPRESA, POBLACIÓN, [FORMA DE PAGO]
FROM CLIENTES INNER JOIN PEDIDOS ON CLIENTES.CÓDIGOCLIENTE = PEDIDOS.[CÓDIGO CLIENTE];
```

EMPRESA	POBLACIÓN	FORMA DE PAGO
BELTRÁN E HIJOS	MADRID	CONTADO
BELTRÁN E HIJOS	MADRID	TARJETA
BELTRÁN E HIJOS	MADRID	CONTADO
BELTRÁN E HIJOS	MADRID	CONTADO
BELTRÁN E HIJOS	MADRID	APLAZADO
BELTRÁN E HIJOS	MADRID	TARJETA
BELTRÁN E HIJOS	MADRID	CONTADO
BELTRÁN E HIJOS	MADRID	CONTADO
BELTRÁN E HIJOS	MADRID	CONTADO
LA MODERNA	OVIEDO	CONTADO
LA MODERNA	OVIEDO	CONTADO
EXPORTASA	MADRID	CONTADO
EXPORTASA	MADRID	CONTADO
EXPORTASA	MADRID	TARJETA
CONFECCIONES AMPARO	GUIJÓN	CONTADO
CONFECCIONES AMPARO	GUIJÓN	CONTADO
LA CASA DEL JUGUETE	MADRID	TARJETA

- Y ejecutamos la consulta basándonos en la tabla **PREVIA**:

```
TRANSFORM COUNT(POBLACIÓN) AS TOTAL_FORMAPAGO
SELECT EMPRESA FROM PREVIA
GROUP BY EMPRESA
PIVOT [FORMA DE PAGO]
```

EMPRESA	APLAZADO	CONTADO	TARJETA
ALMACÉN POPULAR	3	2	
BAZAR EL ARGENTINO	1		5
BAZAR LA FARAONA	3	2	
BELTRÁN E HIJOS	1	6	2
CONFECCIONES AMPARO		2	
CONFECCIONES ARTÍMEZ	1		
CONFECCIONES HERMINIA	1	2	
DEPORTES EL MADRILEÑO		2	
DEPORTES GARCÍA	1		
DEPORTES MORÁN		1	
EL PALACIO DE LA MODA	1		
EXPORTASA		2	1
FERRETERÍA EL CLAVO		1	

SQL (Structured Query Language)

1.3. Hacer consulta para que nos aparezcan dos campos en la zona de filas.

- En este caso sí que nos aparecerán dos registros de tipo DESTORNILLADOR, ya que cada uno es de un país distinto.

```

TRANSFORM SUM(PRECIO) AS TOTAL
SELECT NOMBREARTÍCULO, PAÍSDEORIGEN FROM PRODUCTOS
GROUP BY NOMBREARTÍCULO, PAÍSDEORIGEN
PIVOT SECCIÓN
    
```

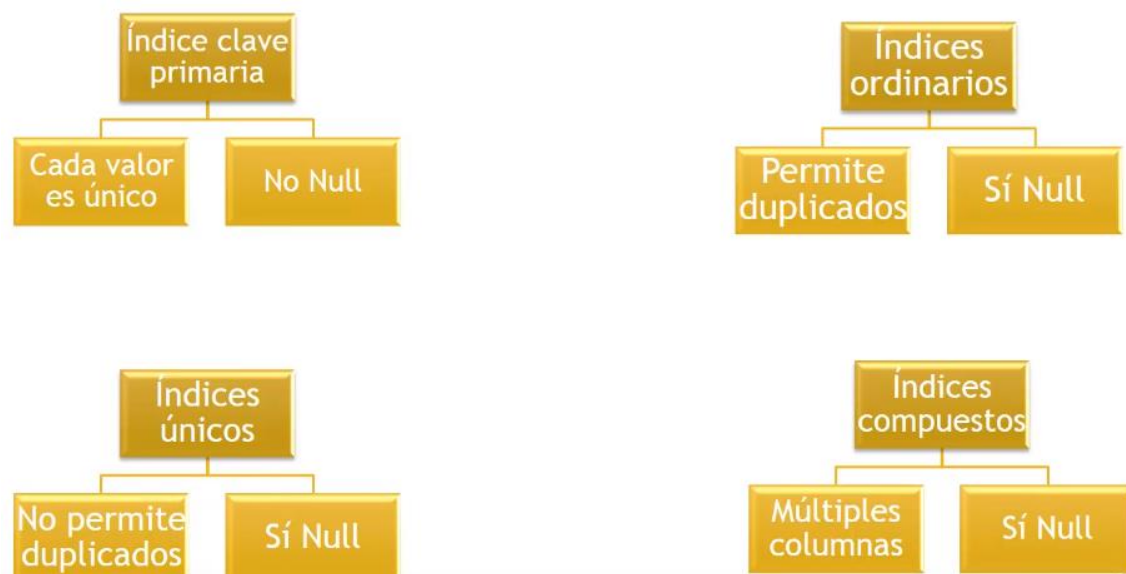
NOMBREARTÍCULO	PAÍSDEORIGEN	CERÁMICA	CONFECCIÓN	DEPORTES	DEPORTIVOS	FERRETERÍA	JUGUETERÍA	OFICINA
ABRIGO CABALLERO	ITALIA	500.000,00 €						
ABRIGO SRA	MARRUECOS		360,07 €					
ALICATES	ITALIA					6,74 €		
BALÓN BALONCESTO	JAPÓN				75,27 €			
BALÓN FÚTBOL	ESPAÑA				43,91 €			
BALÓN RUGBY	USA				111,64 €			
BLUSA SRA.	CHINA		101,06 €					
BOTA ALPINISMO	ESPAÑA				144,00 €			
CAMISA CABALLERO	ESPAÑA		67,13 €					
CAMISETA	ESPAÑA							
CAMISETA2	ESPAÑA							
CAMISETA3	ESPAÑA							
CAÑA DE PESCA	USA				270,00 €			
CAZADORA PIEL	ITALIA		522,69 €					
CENICERO	JAPÓN	19,75 €						
CINTURÓN DE PIEL	ESPAÑA		4,33 €					
COCHE TELEDIRIGIDO	MARRUECOS						159,45 €	
CONSOLA VIDEO	USA						442,54 €	
CORREPASILLOS	JAPÓN						103,34 €	
CRONÓMETRO	USA				439,18 €			
DESTORNILLADOR	ESPAÑA					6,63 €		
DESTORNILLADOR	FRANCIA					9,06 €		
FUERTE DE SOLDADOS	JAPÓN						143,70 €	
JARRA CHINA	CHINA	177,77 €						

Referencias

- Nos permiten hacer búsquedas más rápidas en una tabla de una BBDD.
- No forman parte del estándar de SQL (cada gestor de BBDD tiene su propio comando).

► Creación de índices:

- Índices de clave primaria
- Índices ordinarios
- Índices únicos
- Índices compuestos



- **Indicar CLAVE PRIMARIA al crear una tabla:**

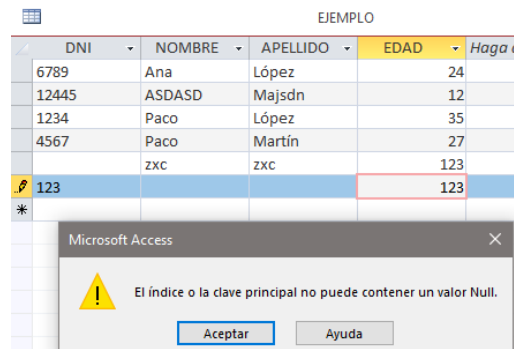
```
CREATE TABLE EJEMPLO (DNI TEXT, NOMBRE TEXT, APELLIDO TEXT, EDAD NUMBER, PRIMARY KEY (DNI))
```

- Indicar clave primaria con la tabla ya creada:

```
ALTER TABLE EJEMPLO ADD PRIMARY KEY (DNI)
```

Índice clave primaria

- No permite dejar un índice vacío y tampoco repetir los dos índices:
- **Dos claves, NOMBRE y APELLIDO (No podemos dejar los índices vacíos y tampoco, podemos repetirlos (No pueden haber 2 Paco López)).**



Índice ordinario

- Permite duplicado y permite dejar en blanco el CAMPO.
- **Crear índice, especificando cuál es el REGISTRO que queremos que tenga el índice:**

`CREATE INDEX MIINDICE ON EJEMPLO (APELLIDO)`

Nombre del campo	Tipo de datos	
DNI	Texto corto	
NOMBRE	Texto corto	
APELLIDO	Texto corto	
EDAD	Número	

Índices: EJEMPLO		
Nombre de índice	Nombre del campo	Criterio de ordenación
MIINDICE	APELLIDO	Ascendente

Índice único

- Permite duplicado y permite dejar en blanco el CAMPO.
- **Crear índice, especificando cuál es el REGISTRO que queremos que tenga el índice:**

`CREATE UNIQUE INDEX MIINDICE ON EJEMPLO (APELLIDO)`

Índice compuesto

- Es el que está compuesto por 2 o más campos y permite dejar en blanco el CAMPO. No permitirá haber 2 personas con el mismo nombre y apellido.
- **Crear índice, especificando cuál es el REGISTRO que queremos que tenga el índice:**

`CREATE UNIQUE INDEX MIINDICE ON EJEMPLO (NOMBRE, APELLIDO)`

Eliminar índice y clave primaria

- **En Acces, SQL Server, Oracle:**

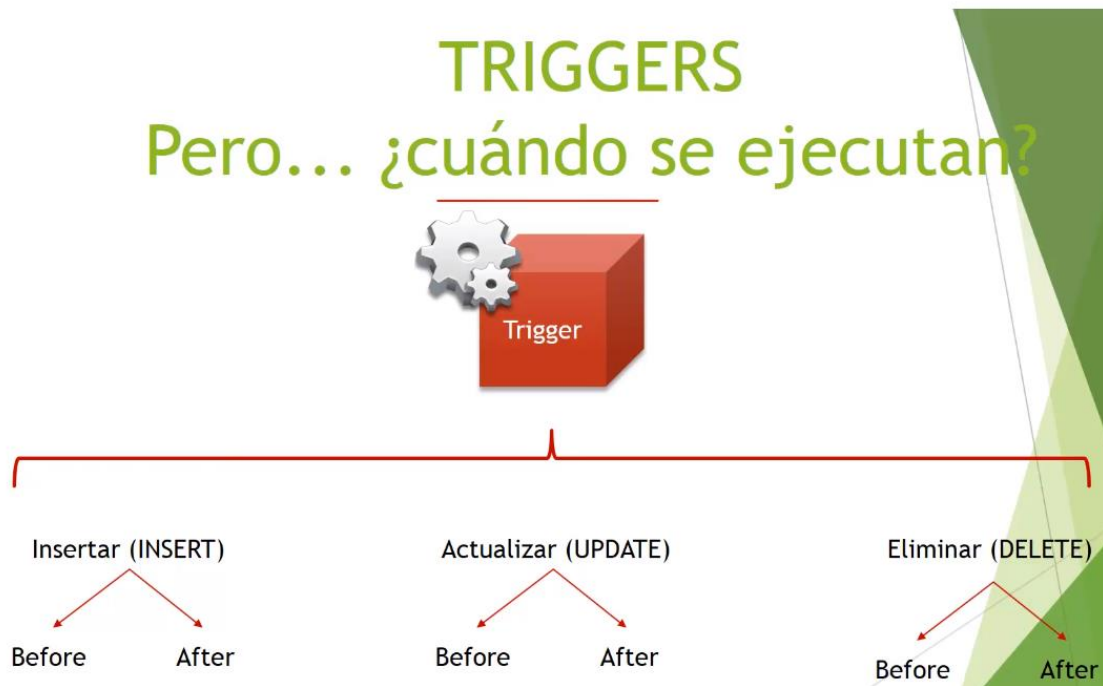
```
ALTER TABLE EJEMPLO DROP CONSTRAINT Index_2E945B32_F3AF_4D42
```

- **En MYSQL:**

```
ALTER TABLE EJEMPLO DROP PRIMARY KEY
```

Triggers (Disparadores)

- Desencadena una acción cuando se hacen modificaciones de tipo INSERT, UPDATE, ELIMINAR.
- En Acces no se pueden utilizar.



SQL (Structured Query Language)

- **(TRIGGER INSERT)** Cuando insertamos un nuevo producto en la tabla **PRODUCTOS**, insertar el mismo producto (especificando los campos a rellenar de la nueva tabla) en otra tabla llamada **REG_PRODUCTOS**.

```
CREATE TRIGGER PRODUCTOS_AI AFTER INSERT ON productos
FOR EACH ROW INSERT INTO reg_productos (CÓDIGOARTÍCULO, NOMBREARTÍCULO, PRECIO, INSERTADO)
VALUES (NEW.CÓDIGOARTÍCULO, NEW.NOMBREARTÍCULO, NEW.PRECIO, NOW());
```

- **FOR EACH ROW:** Por cada fila que insertemos, se inserta otra, en la tabla **reg_productos**.
- **NOW():** Función que crea la fecha y la hora en la que se ha ejecutado la acción (aparecerá: **INSERTADO**)

2018-09-24 11:09:15

- **(TRIGGER UPDATE)** Al hacer una modificación en un campo de la tabla, ejecutar un trigger **BEFORE** para que cree una tabla alternativa de respaldo para saber el nuevo valor del campo actualizado, y el antiguo.

- Se crea la tabla:

```
CREATE TABLE PRODUCTOS_ACTUALIZADOS (ANTERIOR_CÓDIGO VARCHAR(4), ANTERIOR_NOMBREARTÍCULO VARCHAR(25), ANTERIOR_SECCIÓN VARCHAR (15),
ANTERIOR_PRECIO INT(4), ANTERIOR_IMPORTADO VARCHAR(15), ANTERIOR_PAÍSDEORIGEN VARCHAR(15),
ANTERIOR_FECHA DATE, NUEVO_CÓDIGOARTÍCULO VARCHAR(4), NUEVO_NOMBREARTÍCULO VARCHAR(25),
NUEVO_SECCIÓN VARCHAR (15), NUEVO_PRECIO INT(4), NUEVO_IMPORTADO VARCHAR(15),
NUEVO_PAÍSDEORIGEN VARCHAR(15), NUEVO_FECHA DATE, USUARIO VARCHAR(15), F_MODIF DATE)
```

- Se crea el trigger:

```
CREATE TRIGGER ACTUALIZA_PRODUCTOS_BU BEFORE UPDATE ON PRODUCTOS FOR EACH ROW INSERT INTO productos_actualizados
(ANTERIOR_CÓDIGOARTÍCULO, ANTERIOR_FECHA, ANTERIOR_IMPORTADO, ANTERIOR_NOMBREARTÍCULO, ANTERIOR_PAÍSDEORIGEN,
ANTERIOR_PRECIO, ANTERIOR_SECCIÓN,
NUEVO_CÓDIGOARTÍCULO, NUEVO_FECHA, NUEVO_IMPORTADO, NUEVO_NOMBREARTÍCULO, NUEVO_PAÍSDEORIGEN,
NUEVO_PRECIO, NUEVO_SECCIÓN, USUARIO, F_MODIF)
VALUES (OLD.CÓDIGOARTÍCULO, OLD.FECHA, OLD.IMPORTADO, OLD.NOMBREARTÍCULO, OLD.PAÍSDEORIGEN, OLD.PRECIO, OLD.SECCIÓN,
NEW.CÓDIGOARTÍCULO, NEW.FECHA, NEW.IMPORTADO, NEW.NOMBREARTÍCULO, NEW.PAÍSDEORIGEN, NEW.PRECIO, NEW.SECCIÓN,
CURRENT_USER(), NOW())
```

- Se ejecuta el comando para actualizar:

```
UPDATE PRODUCTOS SET PRECIO=PRECIO+20 WHERE CÓDIGOARTÍCULO
```

- Resultado:

SECCIÓN	ANTERIOR_PRECIO	ANTERIOR_IMPORTADO	ANTERIOR_PAÍSDEORIGEN	ANTERIOR_FECHA	NUEVO_CÓDIGOARTÍCULO	NUEVO_NOMBREARTÍCULO	NUEVO_SECCIÓN	NUEVO_PRECIO	NUEVO_IMPORTADO	NUEVO_PAÍSDEORIGEN
3N	30	VERDADERO	FRANCIA	2001-03-23	AR07	SERRUCHO	CONFECCIÓN	50	VERDADERO	FRANCIA

SQL (Structured Query Language)

- **(TRIGGER DELETE)** Cuando se borra una fila en una tabla, hacer una copia de seguridad de esa fila y guardarla en otra tabla (PROD_ELIMINADOS)

- o Se crea la tabla:

```
CREATE TABLE PROD_ELIMINADOS (C_ART VARCHAR(5), NOMBRE VARCHAR(15), SECCIÓN VARCHAR(15), PRECIO INTEGER, PAÍS_ORIGEN VARCHAR(15))
```

- o Se crea el trigger:

```
CREATE TRIGGER ELIMPROD_AD AFTER DELETE ON PRODUCTOS FOR EACH ROW  
INSERT INTO prod_eliminados (C_ART, NOMBRE, PAÍS_ORIGEN, PRECIO, SECCIÓN)  
VALUES  
(OLD.CÓDIGOARTÍCULO, OLD.NOMBREARTÍCULO, OLD.PAÍSDEORIGEN, OLD.PRECIO, OLD.SECCIÓN)
```

- o Se ejecuta la instrucción:

```
DELETE FROM productos WHERE CÓDIGOARTÍCULO = 'AR03'
```

- o Resultado:

C_ART	NOMBRE	SECCIÓN	PRECIO	PAÍS_ORIGEN
AR03	COCHE TELEDIRIGIDO	JUGUETERÍA	159,45	MARRUECOS

Modificar trigger

Editar disparador

Detalles

Nombre del disparador

ELIMPROD_AD

Tabla

productos

Tiempo

AFTER

Evento

DELETE

Definición

```
1 INSERT INTO PROD_ELIMINADOS (C_ART, NOMBRE, PAÍS_ORIGEN, PRECIO, SECCIÓN)
2 VALUES (OLD.CÓDIGOARTÍCULO, OLD.NOMBREARTÍCULO, OLD.PAÍSDEORIGEN,
OLD.PRECIO, OLD.SECCIÓN)
```

Definidor

root@localhost

Continuar

Cerrar

Procedimientos almacenados

- Creamos procedimiento:

```
CREATE PROCEDURE MUESTRA_CLIENTES()

SELECT * FROM CLIENTES WHERE POBLACIÓN='MADRID'
```

- Si lo llamamos, ejecuta lo que le hemos pedido que haga, al procedimiento:

CALL `MUESTRA_CLIENTES`();

Resultados de la ejecución de la rutina 'MUESTRA_CLIENTES'

CÓDIGOCLIENTE	EMPRESA	DIRECCIÓN	POBLACIÓN	TELÉFONO	RESPONSABLE	HISTORIAL
CT34	BAZAR LA FARAONA	CASTILLA Y LEÓN 34	MADRID	915483627	ANGEL SANTAMARÍA	NULL
CT30	BAZAR EL ARGENTINO	ATOCHA 55	MADRID	912495973	ADRIÁN ÁLVAREZ	NULL
CT26	FERRETERÍA LA ESCOBA	ORENSE 7	MADRID	918459346	JOSÉ GARCÍA	NULL
CT23	EL PALACIO DE LA MODA	ORTEGA Y GASSET 129	MADRID	927785235	LAURA CARRASCO	NULL
CT19	CONFECCIONES GALÁN	FUENCARRAL 78	MADRID	913859234	JUAN GARCÍA	NULL
CT14	DEPORTES GARCÍA	GUZMÁN EL BUENO 45	MADRID	913299475	ANA JIMÉNEZ	NULL
CT10	FERETERÍA EL CLAVO	PASEO DE ÁLAMOS 78	MADRID	914354866	MANUEL MENÉNDEZ	NULL
CT07	LA CASA DEL JUGUETE	AMÉRICA 45	MADRID	912649987	ELÍAS PÉREZ	NULL
CT04	EXPORTASA	VALLECAS 34	MADRID	913452378	ELVIRA GÓMEZ	NULL
CT01	BELTRÁN E HIJOS	LAS FUENTES 78	MADRID	914456435	ANGEL MARTÍNEZ	NULL
CT36	JUGUETES EDUCATIVOS SANZ	ORENSE 89	MADRID	916872354	PEDRO IGLESIAS	NULL
CT39	FERRETERÍA LIMA	VALLECAS 45	MADRID	913532785	LUIS GARCÍA	NULL

Rutinas ⓘ

Nombre	Acción	Tipo	Retorna
<input type="checkbox"/> MUESTRA_CLIENTES	Editar Ejecutar Exportar Eliminar	PROCEDURE	

Consultas con procedimientos con pase de parámetros

- Creamos procedimiento:

```
CREATE PROCEDURE ACTUALIZA_PRODUCTOS(N_PRECIO INT, CODIGO VARCHAR(4))

UPDATE PRODUCTOS SET PRECIO=N_PRECIO WHERE CÓDIGOARTÍCULO =CODIGO;
```

- Llamamos al procedimiento pasándole parámetros:

```
CALL ACTUALIZA_PRODUCTOS(60, 'AR22')
```

SQL (Structured Query Language)

- Procedimientos, con delimitadores:

- Creamos el procedimiento, especificando delimitador de bloque (\$\$), y por cada declaración de variable añadimos ';'. Luego especificamos al final que el delimitador de línea será '\$\$'.

```
DELIMITER $$

CREATE PROCEDURE CALCULA_EDAD(ANYO_NACIMIENTO INT)

BEGIN

    DECLARE ANYO_ACTUAL INT DEFAULT 2018;
    DECLARE EDAD INT;
    SET EDAD=ANYO_ACTUAL-ANYO_NACIMIENTO;
    SELECT EDAD;

END;$$

DELIMITER ;
```

- Ejecutamos la consulta y el resultado es correcto:

```
call calcula_edad(1995)
```

EDAD
23

- Procedimientos, con delimitadores y condicionales:

- Crear procedimiento, que al modificar el precio de los productos, aplica un condicional (en el caso que el precio sea menor que 0, no pintar en la tabla un número negativo, y en el caso de que sea superior a 1000, que aparezca solo el número 1000).

```
DELIMITER $$

CREATE TRIGGER REvisa_PRECIO_BU BEFORE UPDATE ON PRODUCTOS FOR EACH ROW

BEGIN

    IF(NEW.PRECIO<0) THEN
        SET NEW.PRECIO=0;
    ELSEIF(NEW.PRECIO>1000) THEN
        SET NEW.PRECIO=1000;
    END IF;

END;$$

DELIMITER ;
```

- Ejecutamos la modificación y el resultado es correcto:

```
UPDATE PRODUCTOS SET PRECIO=15 WHERE CÓDIGOARTÍCULO='AR01'
```

CÓDIGOARTÍCULO	SECCIÓN	NOMBREARTÍCULO	PRECIO	FECHA	IMPORTADO	PAÍSDEORIGEN	FOTO
AR01	FERRETERIA	DESTORNILLADOR	15.00	2000-10-22	FALSO	ESPAÑA	NULL

- Modificación de precio que supera los 1000:

```
UPDATE PRODUCTOS SET PRECIO=8500 WHERE CÓDIGOARTÍCULO='AR01'
```

Vistas

- Privacidad de la información.
- Optimización de la BBDD.
- Entorno de pruebas (para desarrolladores, para comprobar que datos existen en la BBDD, sin peligrar los datos)

- Crear vista para ver campos de la SECCIÓN DEPORTES en la tabla PRODUCTOS:

- Creación VISTA:

```
CREATE VIEW ART_DEPORTES AS
```

```
SELECT NOMBREARTÍCULO, SECCIÓN, PRECIO FROM PRODUCTOS WHERE SECCIÓN= 'DEPORTES'
```

- Resultado:

`SELECT * FROM `art_deportes``

> >> ☐ Mostrar todo | Número de filas: 25 ▼ | Filtrar filas:

+ Opciones

			NOMBREARTÍCULO	SECCIÓN	PRECIO
<input type="checkbox"/>	Editar	Copiar	Borrar	RAQUETA TENIS	DEPORTES 93,47
<input type="checkbox"/>	Editar	Copiar	Borrar	MANCUERNAS	DEPORTES 60,00
<input type="checkbox"/>	Editar	Copiar	Borrar	PISTOLA OLÍMPICA	DEPORTES 46,73
<input type="checkbox"/>	Editar	Copiar	Borrar	BALÓN RUGBY	DEPORTES 111,64
<input type="checkbox"/>	Editar	Copiar	Borrar	BALÓN BALONCESTO	DEPORTES 75,27
<input type="checkbox"/>	Editar	Copiar	Borrar	CRONOMETRO	DEPORTES 439,18
<input type="checkbox"/>	Editar	Copiar	Borrar	CAÑA DE PESCA	DEPORTES 270,00
<input type="checkbox"/>	Editar	Copiar	Borrar	BOTA ALPINISMO	DEPORTES 144,00

- Si se hace modificación en la tabla principal, en la SECCIÓN DEPORTES, también se verá reflejada la modificación en la vista.

- Eliminar VISTA:

```
DROP VIEW ART_DEPORTES;
```

- Modificar VISTA:

```
ALTER VIEW ART_DEPORTES AS
SELECT NOMBREARTÍCULO, SECCIÓN, PAÍSDEORIGEN FROM PRODUCTOS
WHERE PAÍSDEORIGEN
```