

Actividad 7. Consultas simples con SELECT. Funciones de columna y agrupación.

Ejercicios

→ **Ejercicio 1.** Supongamos que la tabla de **facturas** tiene 2000 filas. La tabla de **vendedores** tiene 40 filas.

Como sabes, la columna **facturas.codven** es clave ajena a la tabla **vendedores**: hace referencia al vendedor al que pertenece la factura.

¿Cuántas filas crees que devolverá la siguiente sentencia? Razona la respuesta

```
SELECT codven  
FROM facturas;
```

Devolverá 2000 filas, una por cada factura de la tabla facturas.

¿Cuántas filas crees que devolverá la siguiente sentencia? Razona la respuesta

```
SELECT DISTINCT codven  
FROM facturas;
```

Devolverá un máximo de 41 filas, ya que DISTINCT agrupa las filas que sean iguales. Entonces suponiendo que todos los vendedores tengan alguna factura son 40 y una fila más si hay alguna factura con codven a null.

¿Cuántas filas crees que devolverá la siguiente sentencia? Razona la respuesta

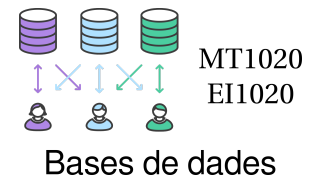
```
SELECT DISTINCT codfac  
FROM facturas;
```

Devolverá 2000 filas, una por cada factura de la tabla facturas, ya que codfac es clave primaria y no estará repetida. En este caso DISTINCT no cambia el resultado de la consulta.

→ **Ejercicio 2.** ¿A qué consulta de datos responde la siguiente sentencia? Contesta en lenguaje natural, como si se lo explicaras a alguien que no sabe nada de SQL

```
SELECT codart, descrip, precio, COALESCE(stock,0) AS stock  
FROM articulos  
ORDER BY stock, descrip;
```

Esta consulta muestra de todos los artículos el código, la descripción, el precio y el stock. En el caso del stock, si no se sabe cual es muestra 0. Ordena los artículos por el stock y después por la descripción.



¿Qué debemos añadir a la sentencia anterior para obtener también el coste almacenado de cada artículo? Recuerda que el stock nos indica cuántas unidades hay de cada artículo.

En la línea de SELECT habría que añadir `COALESCE(stock,0)*precio AS costealmacen`:

`SELECT codart, descrip, precio, COALESCE(stock,0) AS stock, COALESCE(stock,0)*precio AS costealmacen`

`FROM articulos`

`ORDER BY stock, descrip;`

→ **Ejercicio 3.** Los nulos en **facturas.iva** y **facturas.dto** los interpretarlos como el valor 0 (ojo: *ya hemos dicho que es un mal uso, aunque habitual, de los nulos*).

¿Qué debemos escribir en el SELECT de la siguiente sentencia para ver todos los datos de la factura **pero** sin ver los nulos en el iva y el dto (queremos ver los ceros en su lugar)? Fíjate que no aparecen nulos en el ejemplo

codfac	fecha	codven	iva	dto
6078	2019-10-13	355	0	20.00
6371	2021-04-03	355	8	0.00
6396	2021-05-22	355	8	20.00
6132	2022-01-13	355	0	20.00
6133	2022-01-23	355	0	25.00

`SELECT ...`

`FROM facturas;`

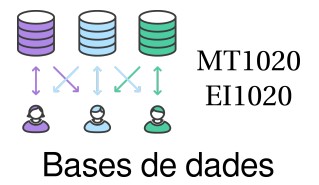
`SELECT codfac, fecha, codven, COALESCE(iva,0) AS iva, COALESCE(dto, 0) AS dto`
`FROM facturas;`

→ **Ejercicio 4.** Mediante **AS** podemos renombrar columnas en el resultado de las consultas.

¿Qué nombre crees que debemos darle a la tercera columna de esta sentencia? Fíjate bien en la función que estamos usando.

`SELECT f.codfac, f.fecha, EXTRACT(doy FROM f.fecha) AS ...dia_año....`
`FROM facturas AS f;`

El nombre debería ser **dia del año** ya que `EXTRACT(doy FROM f.fecha)` (doy son siglas de “day of year”) extrae el día del año de la fecha de la factura.



→ **Ejercicio 5.** Escribe una sentencia SELECT que obtenga como resultado los años en que hay facturas.

```
SELECT DISTINCT EXTRACT(year FROM f.fecha) AS año  
FROM facturas AS f;
```

Agrupación y funciones de columna

Las funciones de columna de SQL son

COUNT(*) Cuenta filas.

COUNT(columna) Cuenta valores no nulos.

COUNT(DISTINCT columna) Cuenta valores distintos y no nulos.

SUM(columna) Suma los valores de la columna (ignora los nulos).

MAX(columna) Obtiene el valor máximo de la columna (ignora los nulos).

MIN(columna) Obtiene el valor mínimo de la columna (ignora los nulos).

AVG(columna) Obtiene la media de los valores de la columna (ignora los nulos).

Suponiendo que tenemos las siguientes facturas en la base de datos

FACTURAS

codfac	fecha	codcli	codven	iva	dto
6371	03/04/2020	129	355	8	0.00
6415	13/06/2020	366	105	8	10.00
6520	03/12/2020	366	405		0.00
5860	07/07/2017		105	18	25.00
5857	05/07/2017		155	8	25.00
6368	31/03/2020	129	305	8	0.00
6380	01/05/2020	342	205		10.00
6088	04/11/2018		305	18	25.00
5768	02/02/2017		405	18	0.00
6085	26/10/2018		155	18	10.00
6328	03/01/2020	12	405	18	10.00
6402	01/06/2020	342	55	18	0.00
6397	25/05/2020	342	255	18	25.00
6404	03/06/2020	171	5		10.00
6500	07/11/2020	171	255	8	
6518	29/11/2020	366	255	18	10.00

(16 filas)

Vas a realizar ejercicios de funciones de columna y agrupación utilizando la tabla de las facturas. teniendo en cuenta que los atributos que **no aceptan nulos son codfac y fecha**.

→ **Ejercicio 6.** Escribe el resultado que se obtiene al ejecutar la siguiente sentencia.

```
SELECT COUNT(*), COUNT(fecha), COUNT(dto), COUNT (iva)
FROM facturas
```

16	16	15	13
----	----	----	----

→ **Ejercicio 7.** Escribe el resultado que se obtiene al ejecutar la siguiente sentencia.

```
SELECT COUNT(DISTINCT codfac), COUNT(codven), COUNT(DISTINCT codven),
COUNT(dto), COUNT(DISTINCT dto)
FROM facturas;
```

16	16	9	15	3
----	----	---	----	---

→ **Ejercicio 8.** Escribe el resultado que se obtiene al ejecutar la siguiente sentencia.

```
SELECT iva, COUNT(*) AS c
FROM facturas
GROUP BY iva;
```

iva	c
8	5
18	8
NULL	3

→ **Ejercicio 9.** Escribe el resultado que se obtiene al ejecutar la siguiente sentencia.

```
SELECT codven, iva, COUNT(*) AS c
FROM facturas
GROUP BY codven, iva;
```

codven	iva	c
5	NULL	1
55	18	1
105	8	1
105	18	1
155	8	1
155	18	1
205	NULL	1
255	8	1
255	18	2
305	8	1
305	18	1
355	8	1
405	NULL	1
405	18	2

→ **Ejercicio 10.** ¿Qué consulta he de hacer para saber qué vendedor tiene al menos 2 facturas en el año 2021?

Si no sabes cómo hacerlo, trata de pensar cómo lo harías si tuvieras un taco de facturas físicas en papel. Te puede ayudar el PDF «[Ejemplo de SELECT con GROUP BY](#)» que hay en el aula virtual.

Para resolver las consultas en SQL puedes seguir estos pasos:

FROM: ¿qué tabla o tablas tienen los datos que me piden en la consulta?

FACTURAS

WHERE: ¿hay que considerar todas las filas de la tabla o sólo algunas? Si hay que considerar sólo algunas necesito un WHERE y el predicado será la condición que han de cumplir dichas filas. Esa condición será una expresión que se evalúe a verdadero o falso.

Solo hay que considerar las del año 2021, necesitaremos un EXTRACT en el WHERE.

GROUP BY: ¿hay que agrupar algunas filas? ¿Necesito hacer alguna operación de columna (contar, máximo, ...) sobre los distintos grupos? En este caso hay que considerar qué columnas o atributos tienen en común los distintos grupos y por esas columnas he de agrupar

Sí, hay que agrupar las del mismo vendedor y contar cuántas tiene.

HAVING: ¿Hay que considerar todos los grupos o hay que eliminar alguno? Si hay que considerar sólo algunos se necesita un HAVING y el predicado será la condición que han de cumplir dichos grupos. Esa condición será una expresión que se evalúe a verdadero o falso y dado que estamos filtrando grupos en la condición obligatoriamente debe aparecer al menos una función de columna

Hay que quedarse con los grupos que tienen 2 o más facturas.

SELECT: ¿qué datos (columnas o expresiones) hay que mostrar de las filas que se han seleccionado?

Se quiere mostrar el codven.

Escribe ahora la consulta

```
SELECT codven
FROM facturas AS f
WHERE EXTRACT(year from f.fecha)=2020
GROUP BY codven
HAVING COUNT (*)>=2;
```