

RESPUESTAS EXAMEN 2º TRIMESTRE DE CONSULTAS SQL 07/04/2025

UNIVERSIDAD_B

1. Devuelve todos los datos del alumno más joven.

```
SELECT *
FROM alumno
WHERE fecha_nacimiento DESC
LIMIT 1;
```

2. Devuelve un listado con todos los alumnos que se han matriculado en alguna asignatura durante el curso escolar 2017/2018.

```
SELECT DISTINCT a.nombre
FROM alumno a
INNER JOIN alumno_se_matricula_asignatura alu ON a.id = alu.id
INNER JOIN curso_escolar c ON alu.id = c.id
WHERE c.anyo_inicio = 2017 AND c.anyo_fin = 2018;
```

3. Devuelve un listado con el nombre de todos los grados existentes en la base de datos y el número total de asignaturas que tiene cada uno. Tenga en cuenta que pueden existir grados que no tienen asignaturas asociadas. Estos grados también tienen que aparecer en el listado. El resultado deberá estar ordenado de mayor a menor por el número de asignaturas.

```
SELECT g.nombre, a.nombre
FROM grado g
INNER JOIN asignatura a ON g.id = a.id
ORDER BY a.nombre ASC;
```

4. Devuelve un listado con el nombre de todos los grados existentes en la base de datos y el número de asignaturas que tiene cada uno, de los grados que tengan más de 40 asignaturas asociadas.

```
SELECT g.nombre, a.nombre
FROM grado g
INNER JOIN asignatura a ON g.id = a.id
GROUP BY g.id, g.nombre
HAVING COUNT(a.id) > 40;
```

JARDINERÍA

5. Devuelve un listado con todos los productos que pertenecen a la gama 'Ornamentales' y que tienen más de 100 unidades en stock. El listado deberá estar ordenado por su precio de venta, mostrando en primer lugar los de mayor precio.

```
SELECT p.nombre  
FROM producto p  
WHERE p.gama = 'Ornamentales'  
AND p.cantidad_en_stock > 100  
ORDER BY p.precio_venta DESC;
```

6. Devuelve el nombre de los clientes y el nombre y apellidos (en una columna) de sus representantes junto con la ciudad de la oficina a la que pertenece el representante.

```
SELECT c.nombre_cliente  
CONCAT (e.nombre, e.apellido1, e.apellido2) AS Representantes, o.ciudad  
FROM cliente c  
INNER JOIN empleado e ON c.codigo_empleado_rep_ventas = e.nombre  
INNER JOIN oficina o ON e.codigo_oficina = o.codigo_oficina;
```

7. Devuelve un listado que muestre solamente los empleados que no tienen un cliente asociado.

```
SELECT e.codigo_empleado, e.nombre  
FROM empleado e  
LEFT JOIN cliente c ON e.codigo_empleado = c.codigo_empleado_rep_ventas  
WHERE c.codigo_empleado IS NULL;
```

8. Devuelve el 'id' de los representantes de ventas y el número de clientes al que atiende cada uno.

```
SELECT c.codigo_empleado COUNT(c.cliente) AS num_cliente  
FROM empleado e  
LEFT JOIN cliente c ON e.codigo_empleado = c.codigo_empleado_rep_ventas  
GROUP BY c.codigo_empleado_rep_ventas;
```

- 9. Calcula la suma de la cantidad total de todos los productos que aparecen en cada uno de los pedidos.**
- 10. Devuelve un listado de los 20 productos más vendidos y el número total de unidades que se han vendido de cada uno. El listado deberá estar ordenado por el número total de unidades vendidas.**
- 11. Devuelve el producto que más unidades tiene en stock.**

```
SELECT p.*  
FROM producto  
ORDER BY p.cantidad_en_stock DESC  
LIMIT 1;
```

- 12. Devuelve un listado de los productos que nunca han aparecido en un pedido.**

```
SELECT p.nombre  
FROM producto p  
WHERE p.codigo_producto NOT IN (  
    SELECT d.codigo_pedido  
    FROM detalle_pedido d  
    WHERE d.codigo_pedido IN(  
        SELECT ped.codigo_pedido  
        FROM pedido ped  
    )  
);
```

- 13. Devuelve un listado que muestre solamente los clientes (id y nombre) que no han realizado ningún pago.**
- 14. Devuelve el listado de clientes donde aparezca el nombre del cliente, el nombre y primer apellido de su representante de ventas y la ciudad donde está su oficina.**

TIENDA_INFORMÁTICA

- 15. Devuelve una lista con el nombre y el precio de todos los productos que contienen la cadena ‘Monitor’ en el nombre y tienen un precio inferior a 215€.**

```
SELECT nombre, precio  
FROM producto  
ORDER BY nombre LIKE '%Monitor%'  
AND precio < 215;
```

- 16. Calcula la media del precio de todos los productos del fabricante ‘Asus’.**

```
SELECT p.nombre, AVG(p.precio) AS media_precio  
FROM producto p  
INNER JOIN fabricante f ON f.id = p.id  
WHERE p.nombre = 'Asus';
```

- 17. Muestra el precio máximo, el precio mínimo, precio medio y el número total de productos de los fabricantes que tienen un precio medio superior a 200€. No es necesario mostrar el nombre del fabricante, con el código del fabricante es suficiente.**

```
SELECT p.nombre,  
MAX(p.precio) AS precio_maximo,  
MIN(p.precio) AS precio_minimo,  
AVG(p.precio) AS media_precio,  
COUNT(p.id) AS total_producto  
FROM producto p  
INNER JOIN fabricante f ON p.id = f.id  
GROUP BY p.id  
HAVING AVG(p.precio) > 200;
```

- 18. Devuelve un listado con los nombres de los fabricantes que tienen 2 o más productos.**

```
SELECT f.nombre, p.nombre  
FROM fabricante f  
INNER JOIN producto p ON f.id = p.id  
GROUP BY f.id, f.nombre  
HAVING COUNT(p.id) >= 2;
```

19. Lista el nombre del producto más barato del fabricante ‘Hewlett-Packard’.

```
SELECT p.nombre
FROM producto p
INNER JOIN fabricante f ON p.id = f.id
WHERE f.nombre = 'Hewlett-Packard'
ORDER BY p.precio ASC
LIMIT 1;
```

20. Devuelve todos los datos de los productos que tienen el mismo precio que el producto más caro del fabricante ‘Lenovo’. (Sin utilizar INNER JOIN).

```
SELECT p.*
FROM producto p
WHERE p.id = (
    SELECT MAX(p.precio)
    FROM producto p
    WHERE p.id = (
        SELECT f.id
        FROM fabricante f
        WHERE f.nombre = 'Lenovo'
    )
);
```