--1.Escribe la sentencia SQL para obtener el nombre de todos los productos (products)

que vende nuestra Empresa:

SELECT name

FROM products;

--2.Obtener los nombres y el precio unitario de los productos (products).

SELECT name, unit\_price

FROM products;

--3. El jefe de compras quiere saber el identificador (id), el nombre y el precio (unitario) de

los productos que tienen un precio inferior o igual a 10,00€.

SELECT product\_id, name

FROM products

WHERE unit\_price <= 10;

--4. El jefe de compras quiere saber el id, el nombre y el precio de los productos que tienen

un valor entre 2,00€ y 99,00€ ambas cantidades incluidas. Escribir 2 sentencias: una

usando AND y otra usando BETWEEN.

SELECT product\_id, name, unit\_price

FROM products

WHERE unit\_price >= 2.00 AND unit\_price <=99.00;

SELECT product\_id, name, unit\_price

FROM products

WHERE unit\_price BETWEEN 2.00 AND 99.00;

--5. Obtener el nombre, el precio, y el precio en dólares ($) de todos los productos,

sabiendo que el precio guardado en la tabla está en euros (€) y que el tipo de cambio está

en: un dólar $1,0 equivale a 0,88€.

SELECT name, unit\_price, unit\_price/0.88 as 'precio\_dolar'

FROM products;

--6. Obtener el número de productos cuyo precio sea mayor o igual a 3,00€.

SELECT COUNT(name)

FROM products

WHERE unit\_price>=3.00;

--7. Obtener el nombre y el precio de los productos cuyo precio sea mayor o igual a 3,00€ y

ordenarlos descendentemente por precio y ascendentemente por nombre.

SELECT name, unit\_price

FROM products

WHERE unit\_price>=3.00

ORDER BY unit\_price DESC, name ASC;

--8. Obtener todos los productos que estén incluidos en una orden de compra (a través de

la tabla order\_items), incluyendo todos los datos del producto, y el identificador de la orden

en el que está incluido. Usa primero inner join explícito y luego implícito.

SELECT p.name,p.unit\_price, o.quantity as "order\_quantity", p.product\_id, o.order\_id

FROM products p

JOIN order\_items o on p.product\_id = o.product\_id;

SELECT p.name,p.unit\_price, o.quantity as "order\_quantity", p.product\_id, o.order\_id

FROM products p

JOIN order\_items o WHERE p.product\_id = o.product\_id;

--9. Obtener los clientes (customers) que hayan nacido después de 1973 y que tengan

más de 200 puntos.

SELECT \*

FROM customers

WHERE birth\_date > '1973-12-31' AND points> 200;

-- 10. Obtener los clientes (customers) que hayan nacido en el estado de Texas (TX),

Florida (FL) o Illinois (IL)

SELECT\*

FROM clients

WHERE state IN ('TX','FL','IL');

-- 11. Obtener los productos cuyo precio unitario esté entre 3,00€ y 29,99€

SELECT\*

FROM products

WHERE unit\_price BETWEEN 3.00 AND 29.99;

-- 12. Obtener los transportistas (shippers) que contengan “ett” o “att” en el nombre. Escribir

dos sentencias, una usando LIKE y la otra con REGEXP

SELECT \*

FROM shippers

WHERE name REGEXP 'ett|att'

-- 13. Obtener los transportistas cuyo nombre acabe en “LLC"

SELECT \*

FROM shippers

WHERE name LIKE '%LLC'

SELECT \*

FROM shippers

WHERE name REGEXP 'LLC$'

-- 14. Obtener el id, fecha de la orden, y comentario de las órdenes de compra (orders) que

tengan comentarios, que el jefe de compras las quiere revisar.

SELECT o.order\_id, order\_date, comments

FROM orders o

JOIN order\_items oi

WHERE comments IS NOT NULL;

-- 15 Obtener las órdenes de compra a partir del 2019 en adelante, que no tengan

comentarios, ordenadas por fecha de manera descendente (la última orden de compra

saldrá la primera).

SELECT\*

FROM orders o

JOIN order\_items oi ON o.order\_id = oi.order\_id

WHERE order\_date >= '2019-01-01' AND comments IS NULL

ORDER BY o.order\_date DESC;

--16. Si un punto equivale a 0,20€, el jefe quiere un listado de clientes (customers) con su

nombre, apellido, puntos y una columna llamada “Cheque regalo” que contendrá el importe

en euros del equivalente a sus puntos, ordenado de mayor cuantía (€) a menos.

SELECT first\_name, last\_name, points, points \* 0.20 AS "Cheque\_regalo\_(€)"

FROM customers

ORDER BY points\*0.20 DESC;

-- 17. Lo mismo que el anterior, pero solo quiere ver los 3 primeros.

SELECT first\_name, last\_name, points, points \* 0.20 AS "Cheque\_regalo\_(€)"

FROM customers

ORDER BY points\*0.20 DESC

LIMIT 3;

-- 18. De la tabla órdenes, obtener id y fecha de la orden, nombre, apellido del comprador/a y

nombre del transportista en caso de haberse enviado la orden.

SELECT o.order\_id, c.first\_name, c.last\_name, s.name

FROM orders o

LEFT JOIN customers c ON o.customer\_id = c.customer\_id

LEFT JOIN shippers s ON o.shipper\_id = s.shipper\_id;

-- 19. De todos los pagos (payments) quiero saber la fecha de pago, número de factura

(invoices), cantidad pagada, nombre y estado del que ha pagado (clients) y el nombre del

método de pago (payment\_methods) con el que ha realizado el pago, en orden, de

manera que vea el último pago realizado el primero de la lista. Se debe de usar USING en

las JOINS explícitas que corresponda.

SELECT p.date AS "Fecha de pago",p.invoice\_id AS "Número de factura", p.amount AS "Cantidad pagada", c.name AS "Nombre del cliente", c.state AS "Estado del cliente", pm.name AS "Nombre del método de pago"

FROM payments p

LEFT JOIN invoices i USING (invoice\_id)

LEFT JOIN clients c ON c.client\_id=i.client\_id

LEFT JOIN payment\_methods pm ON p.payment\_method= pm.name

ORDER BY p.date DESC;

--20. Quiero obtener la siguiente información: de la tabla de facturas (invoices) quiero ver el

id, fecha de factura, total y una columna más que creamos para la ocasión ;) llamada “pago”

cuyo valor será “PAGO PARCIAL” si existe fecha de pago o “SIN COBRAR” si no existe

fecha de pago.

SELECT invoice\_id, invoice\_date AS "Fecha de factura", payment\_total,

IFNULL('PAGO PARCIAL', 'SIN COBRAR') AS pago

FROM invoices;