```
-- ------------------------------------------------------------------------------
-- DDL:
--
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ## -
-- ------------------------------------------------------------------------------




-- ------------------------------------------------------------------------------
-- DDL: ACCIONES BÁSICAS CON TABLAS RELACIONADAS
--      ELIMINAR Y CREAR ÍNDICES Y RESTRICCIONES:
--      SHOW, ALTER TABLE, ADD, DROP, KEY, INDEX, CONSTRAINT, FOREIGN KEY, TRUNCATE
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ## - Eliminar bbdd_pedidos
-- ------------------------------------------------------------------------------
DROP DATABASE bbdd_pedidos;
-- ------------------------------------------------------------------------------
-- ## - Mostar cómo se crearon las tablas de la bbdd_pedidos
-- ------------------------------------------------------------------------------
SHOW CREATE TABLE clientes;
SHOW CREATE TABLE pedidos;
SHOW CREATE TABLE productos_pedidos;
SHOW CREATE TABLE productos;
-- ------------------------------------------------------------------------------
-- ## - Eliminar restricción CONSTRAINT e índice KEY de pedidos con clientes
-- ------------------------------------------------------------------------------
ALTER TABLE pedidos DROP CONSTRAINT fk_pedidos_clientes;
ALTER TABLE pedidos DROP KEY fk_pedidos_clientes;
-- ------------------------------------------------------------------------------
-- ## - Eliminar restricción CONSTRAINT e índice KEY de productos_pedidos con pedidos
-- ------------------------------------------------------------------------------
ALTER TABLE productos_pedidos DROP CONSTRAINT fk_productos_pedidos_pedidos;
ALTER TABLE productos_pedidos DROP KEY fk_productos_pedidos_pedidos;
-- ------------------------------------------------------------------------------
-- ## - Eliminar restricción CONSTRAINT e índice KEY de productos_pedidos con productos
-- ------------------------------------------------------------------------------
ALTER TABLE productos_pedidos DROP CONSTRAINT fk_productos_pedidos_productos;
ALTER TABLE productos_pedidos DROP KEY fk_productos_pedidos_productos;
-- ------------------------------------------------------------------------------
-- ## - Eliminar todos los registros de la tabla pedidos y productos_pedidos
-- ------------------------------------------------------------------------------
TRUNCATE pedidos;
TRUNCATE productos_pedidos;
-- ------------------------------------------------------------------------------
-- ## - Agregar índice y restricción entre la tabla pedidos y clientes
-- ------------------------------------------------------------------------------
ALTER TABLE pedidos
ADD KEY fk_pedidos_clientes (codigo_cliente),
```

```
ADD CONSTRAINT fk_pedidos_clientes
FOREIGN KEY (codigo_cliente)
REFERENCES clientes (codigo_cliente)
ON DELETE CASCADE
ON UPDATE CASCADE;
-- ---------------------------------------------------------------------------------
-- ## - Agregar índice y restricción entre la tabla productos_pedidos y pedidos
-- ---------------------------------------------------------------------------------
ALTER TABLE productos_pedidos
ADD KEY fk_productos_pedidos_pedidos (numero_pedido),
ADD CONSTRAINT fk_productos_pedidos_pedidos
FOREIGN KEY (numero_pedido)
REFERENCES pedidos (numero_pedido)
ON DELETE CASCADE
ON UPDATE CASCADE;
-- ---------------------------------------------------------------------------------
-- ## - Agregar índice y restricción entre la tabla productos_pedidos y productos
-- ---------------------------------------------------------------------------------
ALTER TABLE productos_pedidos
ADD KEY fk_productos_pedidos_productos (codigo_articulo),
ADD CONSTRAINT fk_productos_pedidos_productos
FOREIGN KEY (codigo_articulo)
REFERENCES productos (codigo_articulo)
ON DELETE CASCADE
ON UPDATE CASCADE;
-- ---------------------------------------------------------------------------------
-- ---------------------------------------------------------------------------------
-- ---------------------------------------------------------------------------------
-- ---------------------------------------------------------------------------------




-- ---------------------------------------------------------------------------------
-- CRUD: CREAR (INSERT INTO), CONSULTAR (SELECT), ACTUALIZAR (UPDATE), ELIMINAR (DELETE)
-- ---------------------------------------------------------------------------------
-- ---------------------------------------------------------------------------------
-- ---------------------------------------------------------------------------------
-- ## - CREAR: INSERT INTO, VALUES
-- ---------------------------------------------------------------------------------
INSERT INTO pedidos VALUES (null, 4, '2021-05-15', 'Contado', 3.5, 1);
-- ---------------------------------------------------------------------------------
-- ## - ACTUALIZAR: UPDATE, SET
-- ---------------------------------------------------------------------------------
UPDATE clientes SET
empresa = 'EMPANADAS S.A.',
direccion = 'AVENIDA SIEMPRE VIVA',
poblacion = 'BOGOTÁ',
telefono = '123456789',
responsable = 'PROFE ALBEIRO'
WHERE codigo_cliente = 1;
-- ---------------------------------------------------------------------------------
-- ## - ELIMINAR: DELETE
-- ---------------------------------------------------------------------------------
```

```sql
DELETE FROM clientes WHERE codigo_cliente = 1;
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------




-- ------------------------------------------------------------------------------
-- CONSULTAS GENERALES: *, FROM
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos (*) los campos de la tabla productos
-- ------------------------------------------------------------------------------
SELECT * FROM productos;
-- ------------------------------------------------------------------------------
-- ## - Seleccione los campos sección, nombre_articulo y precio de la tabla productos
-- ------------------------------------------------------------------------------
SELECT seccion, nombre_articulo, precio FROM productos;
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------
-- ------------------------------------------------------------------------------




-- ------------------------------------------------------------------------------
-- CONSULTAS CON CRITERIOS: WHERE
-- ------------------------------------------------------------------------------
-- ## - Seleccione los campos sección, nombre_articulo y precio de la tabla productos
--      donde la sección sea 'CERÁMICA'
-- ------------------------------------------------------------------------------
SELECT seccion, nombre_articulo, precio FROM productos
WHERE seccion = 'CERÁMICA';
-- ------------------------------------------------------------------------------
-- OPERADORES LÓGICOS Y DE COMPARACIÓN
-- LÓGICOS:      AND, OR, NOT
-- COMPARACIÓN:  LIKE, <>, <=, >=, <, >, BEETWEEN, IN, ANY, ALL
-- ------------------------------------------------------------------------------
-- ## - Seleccione los campos sección, nombre_articulo y precio de la tabla productos
--      donde sección sea igual a 'CERÁMICA' y (realmente es 'OR') 'DEPORTES'
-- ------------------------------------------------------------------------------
SELECT seccion, nombre_articulo, precio FROM productos
WHERE seccion = 'CERÁMICA' OR seccion = 'DEPORTES';
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde sección sea igual a
--      'DEPORTES' y su país de origen sea 'USA'
-- ------------------------------------------------------------------------------
SELECT * FROM productos WHERE seccion = 'DEPORTES' AND pais_origen = 'USA';
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde precio sea mayor 300
-- ------------------------------------------------------------------------------
```

```sql
SELECT * FROM productos WHERE precio > 300;
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde la fecha esté entre
--      '2000-03-01' y '2000-04-30'
-- ------------------------------------------------------------------------------
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';
-- ------------------------------------------------------------------------------
-- CONSULTAS ORDENADAS POR UNO O VARIOS CAMPOS: ORDER BY, ASC, DESC
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde la sección sea igual a
--      'CERÁMICA' y 'DEPORTES' y que lo ordene por la sección (Ascendente)
-- ------------------------------------------------------------------------------
SELECT * FROM productos
WHERE seccion = 'CERÁMICA' OR seccion = 'DEPORTES'
ORDER BY seccion ASC;
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde la sección sea igual a
--      'CERÁMICA' y 'DEPORTES' y que lo ordene por la sección (Descendente)
-- ------------------------------------------------------------------------------
SELECT * FROM productos
WHERE seccion = 'CERÁMICA' OR seccion = 'DEPORTES'
ORDER BY seccion DESC;
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde la sección sea igual a
--      'CERÁMICA' y 'DEPORTES' y que lo ordene por el precio
-- ------------------------------------------------------------------------------
SELECT * FROM productos
WHERE seccion = 'CERÁMICA' OR seccion = 'DEPORTES'
ORDER BY precio;
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde la sección sea igual a
--      'CERÁMICA' y 'DEPORTES', después lo ordene por sección y luego por precio
-- ------------------------------------------------------------------------------
SELECT * FROM productos
WHERE seccion = 'CERÁMICA'OR seccion = 'DEPORTES'
ORDER BY seccion, precio;
-- ------------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos donde la sección sea igual a
--      'CERÁMICA' y 'DEPORTES', después lo ordene por sección y país de origen
-- ------------------------------------------------------------------------------
SELECT * FROM productos
WHERE seccion = 'CERÁMICA' OR seccion = 'DEPORTES'
ORDER BY seccion, pais_origen;
-- ------------------------------------------------------------------------------
-- CONSULTAS CALCULADAS: SUM(), AVG(), COUNT(), MAX(), MIN (),
-- GROUP BY, AS (ALIAS), HAVING (POR WHERE),
-- DATE_FORMAT(NOW(),'%Y-%m-%d') AS alias, DATEDIFF(NOW(),fecha)
-- ------------------------------------------------------------------------------
-- ## - Seleccione la sección (agrupación) y sume los precios (cálculo) de la tabla
--      productos y lo agrupe por la sección
-- ------------------------------------------------------------------------------
SELECT seccion, SUM(precio) FROM productos GROUP BY seccion;
-- ------------------------------------------------------------------------------
```

```sql
-- ## - Seleccione la sección (agrupación) y sume los precios (cálculo) de la tabla
--      productos, lo agrupe por la sección y los ordene por precio
-- ------------------------------------------------------------------------------
SELECT seccion, SUM(precio) AS sum_articulos FROM productos
GROUP BY seccion ORDER BY sum_articulos;
-- ------------------------------------------------------------------------------
-- ## - Seleccione la sección (agrupación) y calcule la media de los precios (cálculo)
--      de la tabla productos, lo agrupe por la sección DEPORTES y CONFECCIÓN y los
--      ordene por la media de los artículos
-- ------------------------------------------------------------------------------
SELECT seccion, AVG(precio) AS media_articulos FROM productos
GROUP BY seccion HAVING seccion = 'DEPORTES' OR seccion = 'CONFECCIÓN'
ORDER BY media_articulos;
-- ------------------------------------------------------------------------------
-- ## - Seleccione la población (agrupación) y cuente de los clientes (cálculo) de la
--      tabla clientes, lo agrupe por la población y los ordene descendentemente por
--      la cantidad de clientes
-- ------------------------------------------------------------------------------
SELECT poblacion, COUNT(codigo_cliente) AS num_cliente FROM clientes
GROUP BY poblacion ORDER BY num_cliente DESC
-- ------------------------------------------------------------------------------
-- ## - Seleccione la seccion (agrupación) y calcule el precio más alto (cálculo) de
--      productos, donde la sección sea CONFECCIÓN y los ordene por sección
-- ------------------------------------------------------------------------------
SELECT seccion, MAX(precio) AS precio_alto FROM productos
WHERE seccion = 'CONFECCIÓN' GROUP BY seccion
-- ------------------------------------------------------------------------------
-- ## - Seleccione el articulo, seccion y precio de la tabla productos y cree un campo
--      calculado del precio más el IVA
-- ------------------------------------------------------------------------------
SELECT nombre_articulo, seccion, precio, precio*1.19 FROM productos
-- ------------------------------------------------------------------------------
-- ## - Seleccione el articulo, seccion y precio de la tabla productos y cree un campo
--      calculado del precio más el IVA, llame el nuevo campo como precio_con_iva
-- ------------------------------------------------------------------------------
SELECT nombre_articulo, seccion, precio, precio*1.19 AS precio_con_iva
FROM productos
-- ------------------------------------------------------------------------------
-- ## - Seleccione el articulo, seccion y precio de la tabla productos y cree un campo
--      calculado del precio más el IVA, redondee a dos decimales y llame el nuevo
--      campo como precio_con_iva
-- ------------------------------------------------------------------------------
SELECT nombre_articulo, seccion, precio, ROUND(precio*1.19,2) AS precio_con_iva
FROM productos
-- ------------------------------------------------------------------------------
-- ## - Seleccione el articulo, seccion, precio y fecha de la tabla productos, cree un
--      campo calculado de la diferencia de días entre la fecha almacenada y la fecha
--      actual, agrupelo por la sección DEPORTES
-- ------------------------------------------------------------------------------
SELECT nombre_articulo, seccion, precio, fecha,
DATE_FORMAT(NOW(),'%Y-%m-%d') AS dia_de_hoy, DATEDIFF(NOW(),fecha) AS diferencia_dias
FROM productos WHERE seccion = 'DEPORTES'

-- ## - Seleccione todos los campos de la tabla productos, donde la sección sea
```

```
--      igual a DEPORTES; una el resultado con la selección de todos los campos
--      de la tabla productosnuevos, donde la sección sea igual a DEPORTES DE
--      RIESGO
-- ----------------------------------------------------------------------------
SELECT * FROM productos WHERE seccion = 'DEPORTES' UNION
SELECT * FROM productos_nuevos WHERE seccion = 'DEPORTES DE RIESGO'
-- ----------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos, donde el precio del
--      articulo sea superior a 500 euros y en la tabla productosnuevos,
--      donde la sección sea igual a ALTA COSTURA
-- ----------------------------------------------------------------------------
SELECT * FROM productos WHERE precio > 500 UNION
SELECT * FROM productos_nuevos WHERE seccion = 'ALTA COSTURA'
-- ----------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos, donde la sección sea
--      igual a DEPORTES y en la tabla productosnuevos, todos los productos
--      sin incluir repeticiones
-- ----------------------------------------------------------------------------
SELECT * FROM productos WHERE seccion = 'DEPORTES' UNION
SELECT * FROM productos_nuevos
-- ----------------------------------------------------------------------------
-- ## - Seleccione todos los campos de la tabla productos, donde la sección sea
--      igual a DEPORTES y en la tabla productosnuevos, todos los productos
--      incluyendo repeticiones
-- ----------------------------------------------------------------------------
SELECT * FROM productos WHERE seccion = 'DEPORTES' UNION ALL
SELECT * FROM productos_nuevos
-- ----------------------------------------------------------------------------
-- Inner Join, Outer Joins (Right Join, Left Join [Composiciones Externas])
-- ----------------------------------------------------------------------------
-- ----------------------------------------------------------------------------
-- ----------------------------------------------------------------------------
-- ## - Inner Join: Solo la información común entre las tablas: clientes y
-- pedidos. Clientes de Madrid que SÍ han hecho pedidos
-- ----------------------------------------------------------------------------
SELECT * FROM clientes INNER JOIN pedidos
ON clientes.codigo_cliente = pedidos.codigo_cliente
WHERE poblacion = 'MADRID' ORDER BY clientes.codigo_cliente
-- ----------------------------------------------------------------------------
-- ## - Left Join: La información de la tabla de la izquierda (clientes) y
-- y la información común entre las tablas: clientes y pedidos.
-- Todos los clientes de Madrid y que además hayan hecho pedidos
-- ----------------------------------------------------------------------------
SELECT * FROM clientes LEFT JOIN pedidos
ON clientes.codigo_cliente = pedidos.codigo_cliente
WHERE poblacion = 'MADRID' ORDER BY clientes.codigo_cliente
-- ----------------------------------------------------------------------------
-- ## - Left Join: Ver el codigo_cliente, poblacion, direccion, numero_pedido
-- de la tabla clientes y codigo_cliente, forma_pago de la tabla pedidos donde
-- clientes y pedidos estén relacionados
-- ----------------------------------------------------------------------------
SELECT clientes.codigo_cliente, poblacion, direccion, numero_pedido,
pedidos.codigo_cliente, forma_pago FROM clientes INNER JOIN pedidos
ON clientes.codigo_cliente = pedidos.codigo_cliente
```

```sql
-- ------------------------------------------------------------------------
-- ## - Left Join: Ver el codigo_cliente, poblacion, direccion, numero_pedido
-- de la tabla clientes y codigo_cliente, forma_pago de la tabla pedidos donde
-- clientes y pedidos estén relacionados. Además, filtre solo los de Madrid y
-- los ordene de menor a mayor
-- ------------------------------------------------------------------------
SELECT clientes.codigo_cliente, poblacion, direccion, numero_pedido,
pedidos.codigo_cliente, forma_pago FROM clientes INNER JOIN pedidos
ON clientes.codigo_cliente = pedidos.codigo_cliente
WHERE poblacion = "MADRID" ORDER BY clientes.codigo_cliente
-- ------------------------------------------------------------------------
-- ## - Todos los clientes de Madrid y que no hayan hecho pedidos
-- ------------------------------------------------------------------------
SELECT * FROM clientes LEFT JOIN pedidos
ON clientes.codigo_cliente = pedidos.codigo_cliente
WHERE poblacion = 'MADRID' AND pedidos.codigo_cliente IS NULL
ORDER BY clientes.codigo_cliente
-- ------------------------------------------------------------------------
-- ## - Right Join: La información de la tabla de la derecha (pedidos) y
-- y la información común entre las tablas: clientes y pedidos
-- Todos pedidos que se hayan hecho, así no tengan clientes asociados (OJO)
-- ------------------------------------------------------------------------
SELECT * FROM clientes RIGHT JOIN pedidos
ON clientes.codigo_cliente = pedidos.codigo_cliente
ORDER BY clientes.codigo_cliente
-- ------------------------------------------------------------------------
-- ## - Consultas multitabla: Escalonada, de lista, correlacionada
-- SELECT dentro de otro SELECT
-- ------------------------------------------------------------------------
```