

REALIZACIÓN DE CONSULTAS MULTITABLA CON COMPOSICIÓN INTERNA

1. Base de datos Tienda Informática (2 tablas).

Muestra los nombres de los productos junto con sus fabricantes.

```
SELECT producto.nombre AS nombre_producto, fabricante.nombre AS nombre_fabricante
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo;
```

Muestra todos los datos de ambas tablas juntos.

```
SELECT *
FROM producto INNER JOIN fabricante
ON producto.codigo_fabricante = fabricante.codigo;
```

Muestra todos los datos de ambas tablas a través de una unión natural.

```
SELECT *
FROM producto NATURAL JOIN fabricante;
```

2. Base de datos Gestión de Empleados (2 tablas).

Muestra los empleados y los departamentos donde trabaja cada uno.

```
Select * from empleado inner join departamento on
departamento.codigo = empleado.codigo_departamento;
```

Muestra los códigos de los departamentos y su nombre, pero sólo de los que tienen empleados.

```
select distinct(departamento.codigo), departamento.nombre from departamento inner join empleado on
departamento.codigo = empleado.codigo_departamento;
```

Muestra un listado del código de departamento, su nombre y el valor del presupuesto actual, pero sólo de los que tienen empleados.

```
Select distinct(departamento.codigo), departamento.nombre, (gastos) as 'Presupuesto actual'
from departamento inner join empleado on departamento.codigo = empleado.codigo_departamento;
```

Muestra el nombre del departamento en donde trabaja el empleado Pepe Ruiz Santana.

```
select departamento.nombre from departamento inner join empleado on departamento.codigo = empleado.codigo_departamento and
empleado.nombre = 'Pepe' and empleado.apellido1 = 'Ruiz' and empleado.apellido2 = 'Santana';
```

Muestra los datos de los empleados que trabajan en el departamento de I+D.

```
select * from empleado join departamento on empleado.codigo_departamento = departamento.codigo and departamento.nombre = 'I+D' ;
```

Muestra los datos de los empleados que trabajan los departamentos de Sistemas, Contabilidad o I+D.

```
select * from empleado join departamento on empleado.codigo_departamento = departamento.codigo
and departamento.nombre IN ('Sistemas', 'Contabilidad', 'I+D')
```

Muestra los nombres de los empleados que trabajan en departamentos cuyo presupuesto no esté entre los 100.000 y 200.000€.

```
select empleado.nombre from empleado join departamento on empleado.codigo_departamento = departamento.codigo and
presupuesto NOT BETWEEN 100000 and 200000;
```

Muestra el nombre de los departamentos donde no haya ningún empleado sin segundo apellido (NULL).

```
select distinct(departamento.nombre) from departamento join empleado on departamento.codigo = empleado.codigo_departamento and
empleado.apellido2 IS NULL;
```

3. Base de datos Gestión de Ventas (3 tablas).

Muestra todos los datos de la unión entre las tablas cliente y pedido.

```
SELECT *
FROM cliente INNER JOIN pedido
ON cliente.id = pedido.id_cliente;
```

Muestra todos los datos de la unión entre las tablas comercial y pedido.

```
SELECT *
FROM comercial INNER JOIN pedido
ON comercial.id = pedido.id_comercial;
```

Muestra todos los datos de la unión entre las tablas cliente, comercial y pedido.

```
SELECT *
FROM cliente INNER JOIN pedido ON cliente.id = pedido.id_cliente
INNER JOIN comercial ON pedido.id_comercial = comercial.id;
```

4. Base de datos NBA (4 tablas).

Muestra los datos asociados entre las tablas Jugadores y Equipos.

```
SELECT *
FROM jugadores INNER JOIN equipos
```

REALIZACIÓN DE CONSULTAS MULTITABLA CON COMPOSICIÓN INTERNA

```
ON jugadores.Nombre_equipo = equipo.Nombre;
```

Muestra temporada, nombre jugador y la media de puntos por partido, asistencias y tapones.

```
SELECT estadisticas.temporada, jugadores.Nombre AS nombre_jugador, estadisticas.Puntos_por_partido, estadisticas.Asistencias_por_partido,
       estadisticas.Tapones_por_partido
FROM estadisticas INNER JOIN jugadores
ON estadisticas.jugador = jugadores.codigo;
```

5. Base de datos Departamento Oficinas (5 tablas).

Muestra la unión entre las tablas Categoría y empleado.

```
SELECT empleado.*, categoria.titulo AS categoria_titulo, categoria.salario AS categoria_salario
FROM empleado
JOIN categoria ON empleado.categoria = categoria.categoria;
```

Muestra la unión entre las tablas Dpto_oficinas y oficina.

```
SELECT dpto_oficinas.*, oficina.ciudad AS ciudad_oficina, oficina.region AS region_oficina
FROM dpto_oficinas
JOIN oficina ON dpto_oficinas.oficina = oficina.oficina;
```

6. Base de datos Jardinería (8 tablas).

Muestra los nombres de los clientes y el nombre y apellido de sus representantes de ventas.

```
select nombre_cliente as 'CLIENTE', nombre as 'REPRESENTANTE'
FROM cliente INNER JOIN empleado
ON cliente.codigo_empleado_rep_ventas = empleado.codigo_empleado;
```

Muestra los nombres de los clientes que hayan realizado pagos junto con el nombre de sus representantes de ventas

```
select distinct(nombre_cliente) as 'CLIENTE', nombre as 'REPRESENTANTE'
FROM cliente INNER JOIN empleado INNER JOIN pago
ON cliente.codigo_empleado_rep_ventas = empleado.codigo_empleado AND
cliente.codigo_cliente = pago.codigo_cliente;
```

Muestra los nombres de los clientes que NO hayan realizado pagos, junto con el nombre de sus representantes de ventas.

```
select nombre_cliente as 'CLIENTE', nombre as 'REPRESENTANTE'
FROM cliente INNER JOIN empleado INNER JOIN pago
ON cliente.codigo_empleado_rep_ventas = empleado.codigo_empleado AND
cliente.codigo_cliente NOT IN (select codigo_cliente from pago);
```

Muestra los nombres de los clientes y el nombre y apellido de sus representantes de ventas junto con la ciudad de la oficina a la que pertenece el representante.

```
select nombre_cliente as 'CLIENTE', nombre as 'REPRESENTANTE', oficina.ciudad as 'CIUDAD'
FROM cliente INNER JOIN empleado INNER JOIN oficina
ON cliente.codigo_empleado_rep_ventas = empleado.codigo_empleado AND
empleado.codigo_oficina = oficina.codigo_oficina;
```

Muestra los nombres de los clientes que NO han hecho pagos y el nombre y apellido de sus representantes de ventas junto con la ciudad de la oficina a la que pertenece el representante.

```
select nombre_cliente as 'CLIENTE', nombre as 'REPRESENTANTE', oficina.ciudad as 'CIUDAD'
FROM cliente INNER JOIN empleado INNER JOIN oficina
ON cliente.codigo_empleado_rep_ventas = empleado.codigo_empleado AND
empleado.codigo_oficina = oficina.codigo_oficina AND
cliente.codigo_cliente NOT IN (select codigo_cliente from pago);
```

Muestra la dirección de las oficinas que tengan clientes en Fuenlabrada.

```
select distinct(concat(oficina.linea_direccion1, ' ', oficina.linea_direccion2)), oficina.ciudad
FROM Oficina INNER JOIN empleado INNER JOIN cliente
on oficina.codigo_oficina = empleado.codigo_oficina AND
empleado.codigo_empleado = cliente.codigo_empleado_rep_ventas AND
cliente.ciudad = 'Fuenlabrada';
```

Muestra el nombre de los empleados junto con el nombre de sus jefes.

```
select tabla1.codigo_empleado, concat(tabla1.nombre, ' ', tabla1.apellido1, ' ', tabla1.apellido2) as 'EMPLEADO', tabla1.codigo_jefe,
       concat(tabla2.nombre, ' ', tabla2.apellido1, ' ', tabla2.apellido2) as 'JEFE'
FROM empleado as tabla1 INNER JOIN empleado as tabla2
ON tabla1.codigo_jefe = tabla2.codigo_empleado;
```

Muestra el nombre de los empleados junto con el nombre de sus jefes y el nombre del jefe de cada jefe.

```
select EMP.codigo_empleado, concat(EMP.nombre, ' ', EMP.apellido1, ' ', EMP.apellido2) as 'EMPLEADO', EMP.codigo_jefe,
       concat(JEFE.nombre, ' ', JEFE.apellido1, ' ', JEFE.apellido2) as 'JEFE', JEFE.codigo_jefe,
       concat(JEFE_JEFE.nombre, ' ', JEFE_JEFE.apellido1, ' ', JEFE_JEFE.apellido2) as 'JEFE JEFE'
FROM empleado as EMP INNER JOIN empleado as JEFE INNER JOIN empleado as JEFE_JEFE
ON EMP.codigo_jefe = JEFE.codigo_empleado AND
JEFE.codigo_jefe = JEFE_JEFE.codigo_empleado;
```

Muestra el nombre de los clientes a los que no se les ha entregado a tiempo un pedido.

REALIZACIÓN DE CONSULTAS MULTITABLA CON COMPOSICIÓN INTERNA

```
select nombre_cliente as 'CLIENTE', (fecha_entrega - fecha_esperada) as 'DEMORA'
FROM cliente INNER JOIN pedido
ON cliente.codigo_cliente = pedido.codigo_cliente AND
estado = 'Entregado' AND
fecha_entrega > fecha_esperada;
```

Muestra las diferentes gamas de producto que ha comprado cada cliente.

```
select distinct(nombre_cliente) as 'CLIENTE', gama as 'GAMA'
FROM cliente INNER JOIN detalle_pedido INNER JOIN pedido INNER JOIN producto
ON pedido.codigo_cliente = cliente.codigo_cliente AND
pedido.codigo_pedido = detalle_pedido.codigo_pedido AND
detalle_pedido.codigo_producto = producto.codigo_producto;
```