

MySQL

1-Obtener la información de los pedidos realizados por el Empleado con DNI72850874F.

```
4 • SELECT p.*
5 FROM Pedidos as p
6 INNER JOIN Empleados as e ON (p.cod_empleado = e.cod_empleado)
7 WHERE DNI = '72850874F';
8
```

cod_pedido	fecha_pedido	fecha_envio	fecha_entrega	estado	descripcion	cod_empleado
P_00002	2020-01-01	2020-01-02	NULL	ENVIADO	NULL	3
P_00008	2020-01-05	2020-01-06	2020-01-16	DEVUELTO	NULL	3
P_00013	2020-01-08	NULL	NULL	ESPERA	NULL	3

2-Obtener el nombre, DNI y teléfono de los empleados con pedidos que aún no han sido enviados. El resultset obtenido debe eliminar repeticiones.

```
• SELECT DISTINCT e.nombre, e.DNI, e.tlfno
1 FROM Empleados as e
2 INNER JOIN Pedidos as p ON (e.cod_empleado = p.cod_empleado)
3 WHERE p.estado != 'ENVIADO';
```

nombre	DNI	tlfno
William Wyler	43207030N	555303030
Sydney Pollack	47356203P	555707070
Brian de Palma	16170520W	555606060
Roman Polansky	37615276H	555808080
David Lean	72850874F	555505050

3-Obtener la información de los empleados que no han realizado ningún pedido.

```
SELECT e.*
FROM Empleados as e
LEFT JOIN Pedidos as p ON (e.cod_empleado = p.cod_empleado)
WHERE cod_pedido IS NULL;
```

ult Grid

Filter Rows:

Export:

Wrap Cell Content:

cod_empleado	DNI	Nombre	email	tfno	extension	oficina	departamento	informa_a
1	59840570K	George Roy Hill	georgeroyhill@Metrowoldwynmayer.com	555202020	EXT_01	OF_01	marketing	NULL

4-Obtener la información de los pedidos que no tienen ningún empleado asociado.

```
SELECT p.*
FROM Empleados as e
RIGHT JOIN Pedidos as p ON (e.cod_empleado = p.cod_empleado)
WHERE p.cod_empleado IS NULL;
```

ult Grid

Filter Rows:

Export:


Wrap Cell Content:


cod_pedido	fecha_pedido	fecha_envio	fecha_entrega	estado	descripcion	cod_empleado
------------	--------------	-------------	---------------	--------	-------------	--------------

5-Obtener la información del pago relacionado en el pedido que tiene como identificadorP_00002.

```
SELECT pag.*
FROM Pagos as pag
INNER JOIN Pedidos as ped ON (pag.cod_pedido = ped.cod_pedido)
WHERE ped.cod_pedido = 'P_00002';
```


ult Grid






Filter Rows:

Export:



Wrap Cell Content:



cod_pago	cod_cliente	cantidad	num_plazos	num_pago	cuantia_plazo	cod_pedido	pago_master
2	1	1000	2	1	500	P_00002	NULL

6-Obtener la información de los pedidos sin pagos asociados.

```
• SELECT ped.*  
  FROM Pedidos as ped  
 LEFT JOIN Pagos as pag ON (ped.cod_pedido = pag.cod_pedido)  
 WHERE cod_pago IS NULL;
```

sult Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:						
cod_pedido	fecha_pedido	fecha_envio	fecha_entrega	estado	descripcion	cod_empleado
P_00011	2020-01-07	2020-01-08	NULL	ENVIADO	NULL	6
P_00012	2020-01-07	2020-01-08	NULL	ENVIADO	NULL	7
P_00013	2020-01-08	NULL	NULL	ESPERA	NULL	3
P_00014	2020-01-08	NULL	NULL	ESPERA	NULL	2
P_00015	2020-01-08	NULL	NULL	ESPERA	NULL	2
P_00016	2020-01-09	NULL	NULL	ESPERA	NULL	2

7-Obtener el nombre de los empleados con pedidos con pagos asociados. El resultset obtenido debe eliminar repeticiones, además los nombres se deben mostrar de ordenados alfabéticamente de manera descendente.

```
• SELECT DISTINCT e.nombre  
  FROM Empleados as e  
 INNER JOIN Pedidos as p ON (e.cod_empleado = p.cod_empleado)  
 INNER JOIN Pagos as pag ON (p.cod_pedido = pag.cod_pedido)  
 WHERE pag.cod_pago IS NOT NULL  
 ORDER BY e.nombre DESC;
```

sult Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:	
nombre	
William Wyler	
Sydney Pollack	
Roman Polansky	
David Lean	
Cecil B. DeMille	
Brian de Palma	

8-Obtener el nombre de los empleados con pedidos que no tienen pagos asociados. El resultset obtenido debe eliminar repeticiones, además los nombres se deben mostrar de ordenados alfabéticamente de manera ascendente.

```
• SELECT DISTINCT e.nombre
  FROM Empleados as e
 INNER JOIN Pedidos as ped ON (e.cod_empleado = ped.cod_empleado)
 LEFT JOIN Pagos as pag ON (ped.cod_pedido = pag.cod_pedido)
 WHERE pag.cod_pago IS NULL
 ORDER BY e.nombre ASC;
```

ult Grid			Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
nombre					
Brian de Palma					
David Lean					
Roman Polansky					
William Wyler					

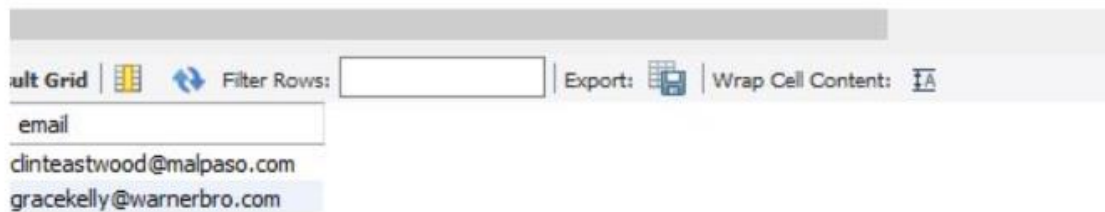
9-Obtener los identificadores de los pedidos con sus pagos asociados (sólo el identificador de los pagos) para el cliente con DNI21492542P.

```
• SELECT ped.cod_pedido, pag.cod_pago
  FROM Pedidos as ped
 INNER JOIN Pagos as pag ON (ped.cod_pedido = pag.cod_pedido)
 INNER JOIN Clientes as c ON (pag.cod_cliente = c.cod_cliente)
 WHERE c.DNI = '21492542P';
```

ult Grid			Filter Rows: <input type="text"/>	Export:	Wrap Cell Content:
cod_pedido		cod_pago			
P_00001		1			
P_00002		2			

10-Obtener la dirección de correo electrónico de los clientes que han comprado el tipo de producto con identificador20.

```
• SELECT c.email
FROM Clientes as c
INNER JOIN Pagos as pag ON (c.cod_cliente = pag.cod_cliente)
INNER JOIN Pedidos as ped ON (pag.cod_pedido = ped.cod_pedido)
INNER JOIN Detalles_pedidos as dp ON (ped.cod_pedido = dp.cod_pedido)
INNER JOIN Productos as pro ON (dp.cod_producto = pro.cod_producto)
WHERE pro.cod_producto = 20;
```

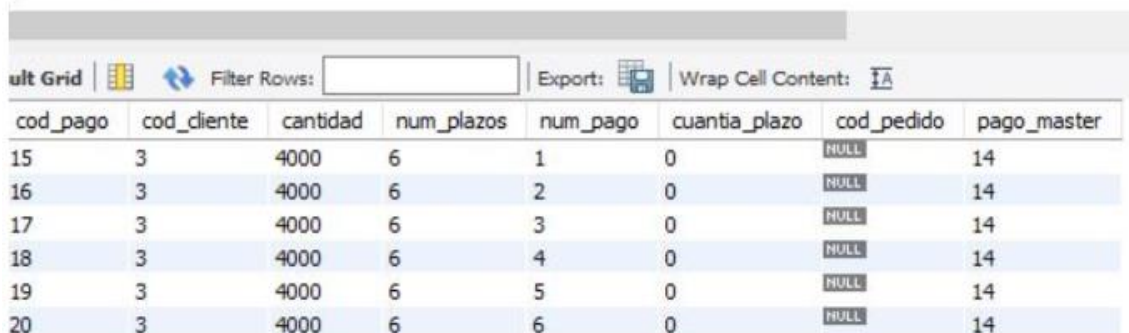


The screenshot shows a database query result in a grid interface. The interface includes a toolbar with icons for 'ult Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar, the column 'email' is selected, and the results are displayed in a list: 'clinteastwood@malpaso.com' and 'gracekelly@warnerbro.com'.

email
clinteastwood@malpaso.com
gracekelly@warnerbro.com

11-Obtener los pagos aplazados del pago maestro con identificador 14.

```
• SELECT pa.*
FROM Pagos as pa
INNER JOIN Pagos pm ON (pa.pago_master = pm.cod_pago)
WHERE pa.pago_master = 14;
```



The screenshot shows a database query result in a grid interface. The interface includes a toolbar with icons for 'ult Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar, a table of payment details is displayed. The table has columns: 'cod_pago', 'cod_cliente', 'cantidad', 'num_plazos', 'num_pago', 'cuantia_plazo', 'cod_pedido', and 'pago_master'. The results show 6 rows of data, all with 'pago_master' value 14.

cod_pago	cod_cliente	cantidad	num_plazos	num_pago	cuantia_plazo	cod_pedido	pago_master
15	3	4000	6	1	0	NULL	14
16	3	4000	6	2	0	NULL	14
17	3	4000	6	3	0	NULL	14
18	3	4000	6	4	0	NULL	14
19	3	4000	6	5	0	NULL	14
20	3	4000	6	6	0	NULL	14

12-Obtener los pagos aplazados del pedidoP_00005.

```
SELECT pa.*
FROM Pagos as pa
INNER JOIN Pagos as pm ON (pa.pago_master = pm.cod_pago)
INNER JOIN Pedidos as ped ON (pm.cod_pedido = ped.cod_pedido)
WHERE ped.cod_pedido = 'P_00005';
```

ult Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:							
cod_pago	cod_cliente	cantidad	num_plazos	num_pago	cuantia_plazo	cod_pedido	pago_master
10	3	2100.025	4	1	0	NULL	9
11	3	2100.025	4	2	0	NULL	9
12	3	2100.025	4	3	0	NULL	9
13	3	2100.025	4	4	0	NULL	9