

```

169
170 -- Obtener los identificadores de los pedidos con sus pagos asociados (sólo el identificador de los pagos)
171
172 • SELECT cod_pago, cod_pedido
173 FROM Pagos pa
174 INNER JOIN Clientes c ON c.cod_cliente = pa.cod_cliente
175 WHERE c.DNI = '21492542P';
176

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	cod_pago	cod_pedido
▶	1	P_00001
	2	P_00002
	5	NULL
	6	NULL

```

177 -- Obtener la dirección de correo electrónico de los clientes que han comprado el tipo de producto con identificador 20.
178 • SELECT email
179 FROM Clientes c
180 INNER JOIN Pagos pa ON pa.cod_cliente = c.cod_cliente
181 INNER JOIN Pedidos pe ON pe.cod_pedido = pa.cod_pedido
182 INNER JOIN Detalles_pedidos dp ON dp.cod_pedido = pe.cod_pedido
183 WHERE dp.cod_producto = '20';
184

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	email
▶	clinteastwood@malpaso.com
	gracekelly@warnerbro.com

```

185 -- Obtener los pagos aplazados del pago maestro con identificador 14.
186 • SELECT num_pago, cod_pago, cantidad
187 FROM Pagos pa
188 WHERE num_pago <> '1'
189 AND pago_master = '14';
190

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	num_pago	cod_pago	cantidad
▶	2	16	4000
	3	17	4000
	4	18	4000
	5	19	4000
	6	20	4000

```

1 -- Preguntar mañana? Una vez el pago se divide, pierden el cod_pedido y ganan como identificador cod_master, sin conocer ese cod_master
2 -- no puedo saber los pagos aplazados
3 • SELECT num_pago, cod_pago, cantidad
4 FROM Pagos pa
5 INNER JOIN Pedidos pe ON pa.cod_pedido = pe.cod_pedido
6 WHERE pa.num_pago <> '1'
7 AND pa.cod_pedido = 'P_00005';
8
9
10

```

ut

Action Output

Time	Action	Message
------	--------	---------

No se hacerlo

```

111 -- EJERCICIO RESUELTO --
112
113
114 -- Pedidos del empleado '72850874F'
115 • SELECT p.*
116 FROM Pedidos p
117 INNER JOIN Empleados e on p.cod_empleado = e.cod_empleado
118 WHERE e.DNI = '72850874F';
119
120
121

```

Result Grid							
Filter Rows: <input type="text"/>							
Export: Wrap Cell Content:							
	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

```

120 -- Obtener el nombre, DNI y teléfono de los empleados con pedidos que aún no han sido enviados.
121
122 • SELECT DISTINCT Nombre, DNI, tfno
123 FROM Empleados e
124 INNER JOIN Pedidos p on e.cod_empleado = p.cod_empleado
125 WHERE p.estado = 'ESPERA';
126
127

```

Result Grid			
Filter Rows: <input type="text"/>			
Export: Wrap Cell Content:			
	Nombre	DNI	tfno
▶	David Lean	72850874F	555505050
	William Wyler	43207030N	555303030



```

127 -- Obtener la información de los empleados que no han realizado ningún pedido
128
129 • SELECT e.*
130 FROM Pedidos p
131 RIGHT JOIN Empleados e ON e.cod_empleado = p.cod_empleado
132 WHERE p.cod_pedido IS NULL;
133

```

Result Grid									
Filter Rows: <input type="text"/>									
Export: Wrap Cell Content:									
	cod_empleado	DNI	Nombre	email	tfno	extension	oficina	departamento	informa_a
▶	1	59840570K	George Roy Hill	georgeroymhill@Metrowoldwynmayer.com	555202020	EXT_01	OF_01	marketing	NULL

```

134 -- Obtener la información de los pedidos que no tienen ningún empleado asociado.
135
136 • SELECT p.*
137 FROM Empleados e
138 INNER JOIN Pedidos p ON p.cod_empleado = e.cod_empleado
139 WHERE p.cod_empleado IS NULL;
140

```

<

Result Grid  Filter Rows: Export:  Wrap Cell Content: 




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

```

141 -- Obtener la información del pago relacionado en el pedido que tiene como identificador P_00002
142 • SELECT pa.*
143 FROM Pedidos pe
144 INNER JOIN Pagos pa ON pa.cod_pedido = pe.cod_pedido
145 WHERE pe.cod_pedido = 'P_00002';
146

```

<

Result 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

```

147 -- Obtener la información de los pedidos sin pagos asociados.
148 • SELECT pe.*
149 FROM Pagos pa
150 RIGHT JOIN Pedidos pe ON pe.cod_pedido = pa.cod_pedido
151 WHERE pa.cod_pago IS NULL;
152

```

<

Result Grid  Filter Rows: 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

```

152
153 -- Obtener el nombre de los empleados con pedidos con pagos asociados
154 • SELECT DISTINCT Nombre
155 FROM Empleados e
156 INNER JOIN Pedidos p ON p.cod_empleado = e.cod_empleado
157 INNER JOIN Pagos pa ON pa.cod_pedido = p.cod_pedido
158 WHERE pa.cod_pago IS NOT NULL
159 ORDER BY Nombre ASC;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Nombre			
▶	Brian de Palma			
	Cecil B. DeMille			
	David Lean			
	Roman Polansky			
	Sydney Pollack			
	William Wyler			

```

161 -- Obtener el nombre de los empleados con pedidos que no tienen pagos asociados.
162
163 • SELECT DISTINCT Nombre
164 FROM Empleados e
165 INNER JOIN Pedidos p ON p.cod_empleado = e.cod_empleado
166 LEFT JOIN Pagos pa ON p.cod_pedido = pa.cod_pedido
167 WHERE pa.cod_pago IS NULL
168 ORDER BY Nombre DESC;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Nombre			
▶	William Wyler			
	Roman Polansky			
	David Lean			
	Brian de Palma			