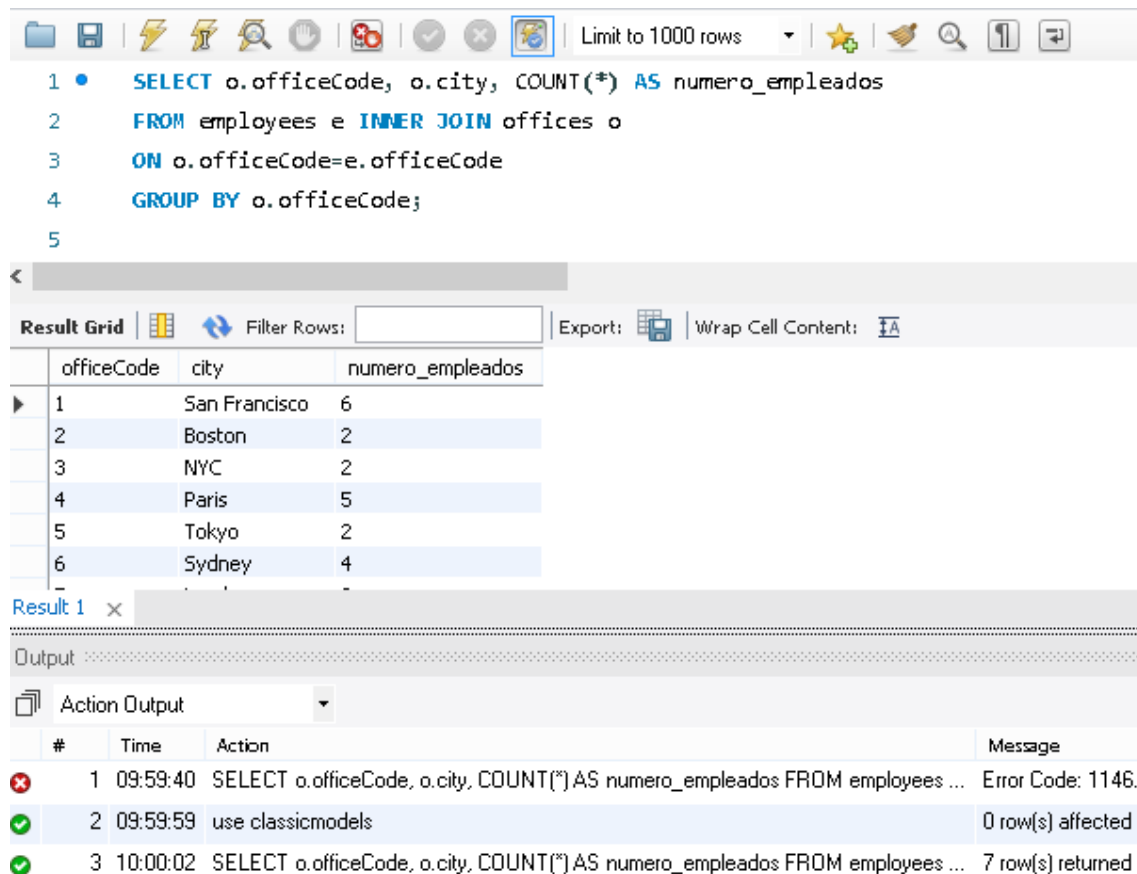


**1-Mostrar los datos de todos los clientes (customers) y de los empleados (employees) que tienen asignados. En caso de que haya algún cliente que no tenga ningún empleado asignado también deberemos obtener los datos de los clientes.**

```
SELECT c.customerName, c.customerNumber, e.firstName, e.employeeNumber FROM  
customers c INNER JOIN employees e ON e.employeeNumber=c.salesRepEmployeeNumber
```

**2-Mostrar el número de empleados (employees) por oficina (office) de mayor a menor. Para la oficina basta con mostrar el código y la ciudad (city) en la que está situada.**

```
SELECT o.officeCode, o.city, COUNT(*) AS numero_empleados  
  
FROM employees e INNER JOIN offices o  
  
ON o.officeCode=e.officeCode  
  
GROUP BY o.officeCode;
```



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL query is displayed in a text editor. The query is as follows:

```
1 • SELECT o.officeCode, o.city, COUNT(*) AS numero_empleados  
2 FROM employees e INNER JOIN offices o  
3 ON o.officeCode=e.officeCode  
4 GROUP BY o.officeCode;  
5
```

Below the query editor, there is a "Result Grid" section. It contains a table with the following data:

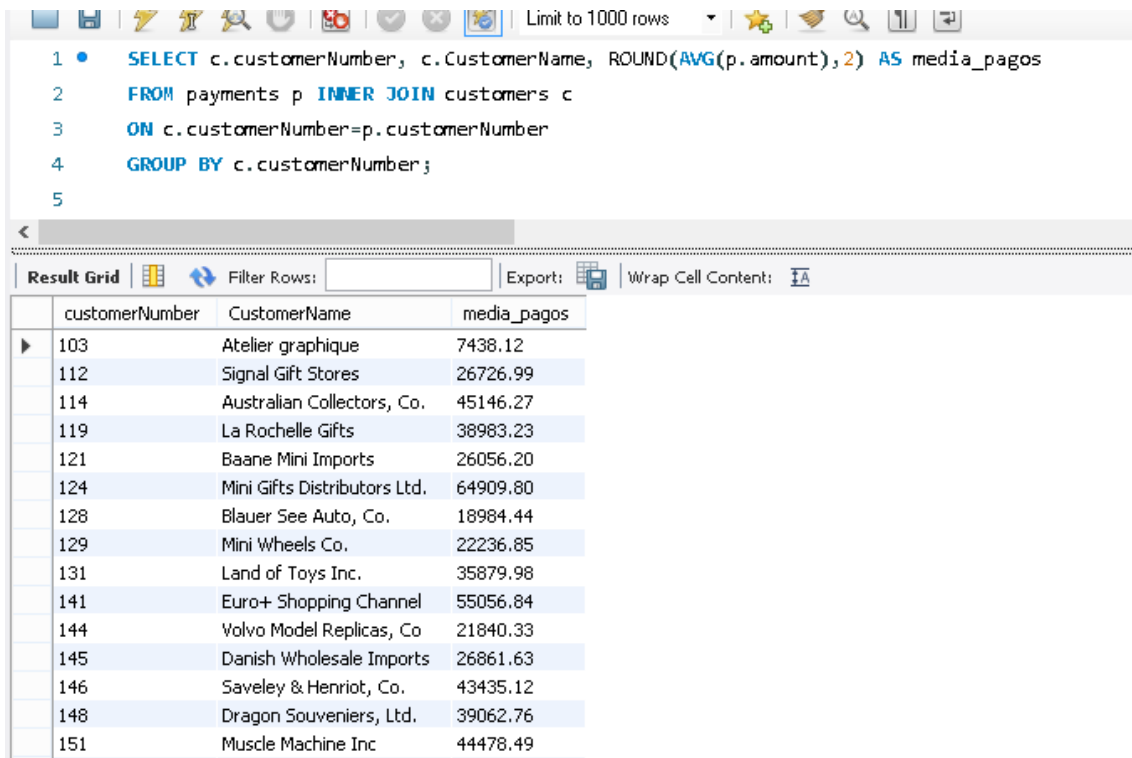
	officeCode	city	numero_empleados
1	San Francisco	6	
2	Boston	2	
3	NYC	2	
4	Paris	5	
5	Tokyo	2	
6	Sydney	4	

Below the result grid, there is an "Output" section. It contains a table with the following data:

#	Time	Action	Message
1	09:59:40	SELECT o.officeCode, o.city, COUNT(*) AS numero_empleados FROM employees ...	Error Code: 1146.
2	09:59:59	use classicmodels	0 row(s) affected
3	10:00:02	SELECT o.officeCode, o.city, COUNT(*) AS numero_empleados FROM employees ...	7 row(s) returned

**3-Mostrar los datos de los clientes (customers) y la media de los pagos (payments) que han realizado cada uno de ellos. Debes considerar el campo amount para la media. La media debe tener 2 decimales.**

```
SELECT c.customerNumber, c.CustomerName, ROUND(AVG(p.amount),2) AS media_pagos
FROM payments p INNER JOIN customers c
ON c.customerNumber=p.customerNumber
GROUP BY c.customerNumber;
```

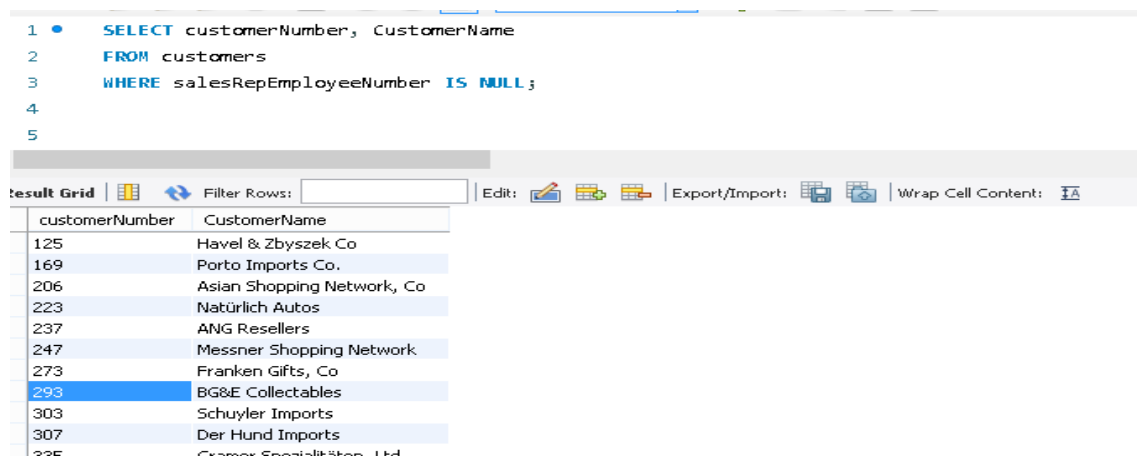


The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, editing, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the query from the previous block. Below the editor is a 'Result Grid' tab with a table of results. The table has three columns: 'customerNumber', 'CustomerName', and 'media\_pagos'. It lists 15 customers with their respective average payment amounts, rounded to two decimal places.

customerNumber	CustomerName	media_pagos
103	Atelier graphique	7438.12
112	Signal Gift Stores	26726.99
114	Australian Collectors, Co.	45146.27
119	La Rochelle Gifts	38983.23
121	Baane Mini Imports	26056.20
124	Mini Gifts Distributors Ltd.	64909.80
128	Blauer See Auto, Co.	18984.44
129	Mini Wheels Co.	22236.85
131	Land of Toys Inc.	35879.98
141	Euro+ Shopping Channel	55056.84
144	Volvo Model Replicas, Co	21840.33
145	Danish Wholesale Imports	26861.63
146	Saveley & Henriot, Co.	43435.12
148	Dragon Souvenirs, Ltd.	39062.76
151	Muscle Machine Inc	44478.49

**4-Mostrar los clientes (customers) que no han realizado ningún pago (payments).**

```
SELECT customerNumber, CustomerName FROM customers WHERE
salesRepEmployeeNumber IS NULL;
```

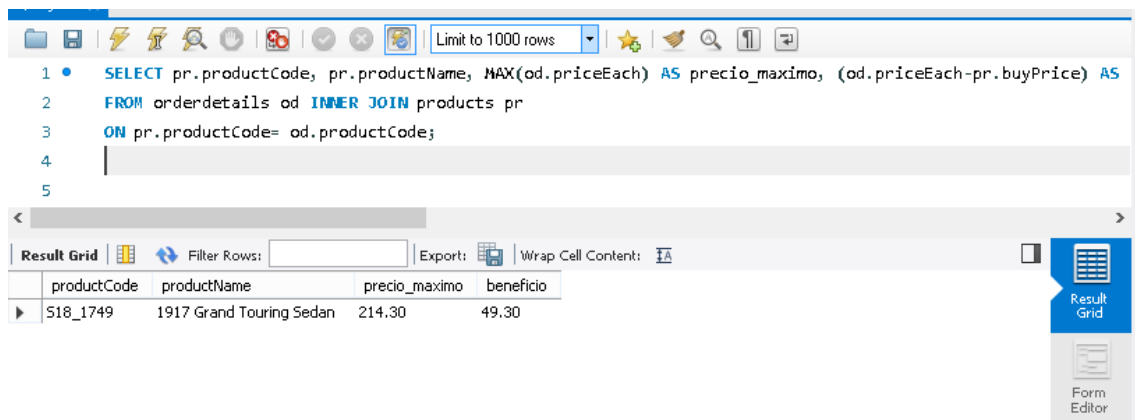


The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, editing, and an 'Export/Import' button. The SQL editor contains the query from the previous block. Below the editor is a 'Result Grid' tab with a table of results. The table has two columns: 'customerNumber' and 'CustomerName'. It lists 10 customers whose sales representative employee number is NULL.

customerNumber	CustomerName
125	Havel & Zbyszek Co
169	Porto Imports Co.
206	Asian Shopping Network, Co
223	Natürlich Autos
237	ANG Resellers
247	Messner Shopping Network
273	Franken Gifts, Co
293	BG&E Collectables
303	Schuyler Imports
307	Der Hund Imports

**5-Mostrar el coche más caro que se ha vendido (products). Considera priceEach como el precio de venta. Muestra además del nombre del coche, el beneficio que ha tenido la empresa en dicha venta.**

```
SELECT pr.productCode, pr.productName, MAX(od.priceEach) AS precio_maximo,  
(od.priceEach-pr.buyPrice) AS beneficio  
  
FROM orderdetails od INNER JOIN products pr  
  
ON pr.productCode= od.productCode;
```

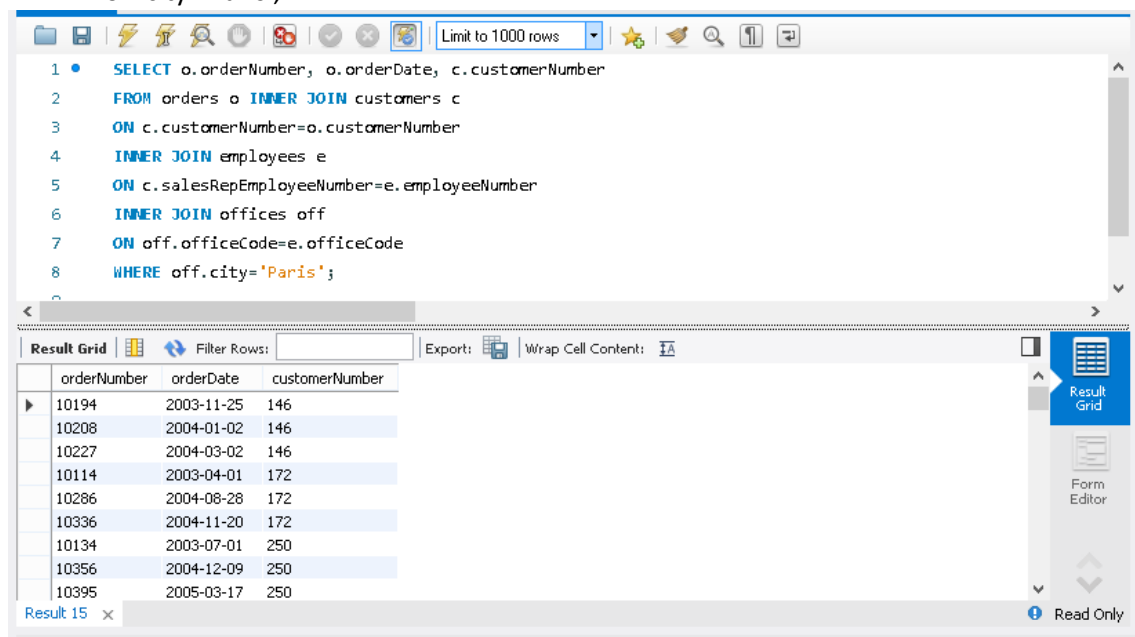


The screenshot shows a SQL IDE with a query editor and a result grid. The query is the same as the one above. The result grid shows one row of data.

productCode	productName	precio_maximo	beneficio
S18_1749	1917 Grand Touring Sedan	214.30	49.30

**6-Mostrar los datos de los pedidos(orders) realizados por clientes(customers) que estén atendidos por empleados(employees) de la oficina (offices) de París.**

```
SELECT o.orderNumber, o.orderDate, c.customerNumber FROM orders o INNER JOIN  
customers c ON c.customerNumber=o.customerNumber INNER JOIN employees e  
ON c.salesRepEmployeeNumber=e.employeeNumber INNER JOIN offices off  
ON off.officeCode=e.officeCode  
  
WHERE off.city='Paris';
```



The screenshot shows a SQL IDE with a query editor and a result grid. The query is the same as the one above. The result grid shows 15 rows of data.

orderNumber	orderDate	customerNumber
10194	2003-11-25	146
10208	2004-01-02	146
10227	2004-03-02	146
10114	2003-04-01	172
10286	2004-08-28	172
10336	2004-11-20	172
10134	2003-07-01	250
10356	2004-12-09	250
10395	2005-03-17	250

**7-Mostrar el total (usar el priceEach como el precio de venta) de cada pedido (orders) del año 2004. Se deberá añadir el símbolo \$ al final del coste.**

```
SELECT od.orderNumber AS NUM_PEDIDO, YEAR(o.orderDate) AS ANIO,  
CONCAT(ROUND(SUM(od.priceEach*od.quantityOrdered),2),'$') AS TOTAL FROM orderdetails od  
INNER JOIN orders o  
ON od.orderNumber=o.orderNumber  
WHERE YEAR(o.orderDate)=2004;
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT od.orderNumber AS NUM_PEDIDO, YEAR(o.orderDate) AS ANIO,  
2 CONCAT(ROUND(SUM(od.priceEach*od.quantityOrdered),2),'$') AS TOTAL FROM orderdetails od  
3 INNER JOIN orders o  
4 ON od.orderNumber=o.orderNumber  
5 WHERE YEAR(o.orderDate)=2004;  
6
```

Below the editor is the 'Result Grid' showing the results of the query:

NUM_PEDIDO	ANIO	TOTAL
10208	2004	4515905.51\$

**8-Mostrar el mes (en letras), año y la cantidad de pedidos que ha habido cada mes desde Enero del 2003 hasta Diciembre de 2005 (considerar el campo orderDate). Ordenar cronológicamente.**

```
SELECT monthname(o.orderDate) AS MES, YEAR(o.orderDate) AS ANIO,  
COUNT(o.orderNumber) AS NUM_PEDIDOS FROM orders o WHERE ((MONTH(o.orderDate)>=1  
AND YEAR(o.orderDate)>=2003) AND (MONTH(o.orderDate)<=12 AND  
YEAR(o.orderDate)<=2005)) GROUP BY MONTH(o.orderDate), YEAR(o.orderDate) ORDER BY  
o.orderDate;
```

The screenshot shows a SQL query editor with the following query:

```
1 SELECT monthname(o.orderDate) AS MES, YEAR(o.orderDate) AS ANIO, COUNT(o.orderNumber) AS NUM_PEDIDOS  
2 FROM orders o  
3 WHERE ((MONTH(o.orderDate)>=1 AND YEAR(o.orderDate)>=2003) AND (MONTH(o.orderDate)<=12 AND YEAR(o.orderDate)<=2005))  
4 GROUP BY MONTH(o.orderDate), YEAR(o.orderDate)  
5 ORDER BY o.orderDate;  
6
```

Below the editor is the 'Result Grid' showing the results of the query:

MES	ANIO	NUM_PEDIDOS
January	2003	5
February	2003	3
March	2003	6
April	2003	7
May	2003	6
June	2003	7
July	2003	7
August	2003	5
September	2003	8

**9-Muestra el apellido (lastame) del empleado (employees) que ha participado en más pedidos(orders). Se entiende que un empleado participa en un pedido si lo hace un cliente (customer) que tiene a su cargo. En caso de empate deben salir todos los empleados empatados en cabeza (no usar limit.)**

**10-Hallar la diferencia entre el producto (products) mas caro y el más barato. Considerar que los precios (buyprice) se encuentran en miles de dólares, por lo que el resultado hay que multiplicarlo por 1000. La columna se llamará Diferencia y tendrá un \$ al final.**

```
SELECT MAX(pr.buyPrice) AS MAXIMO, MIN(pr.buyPrice) AS MINIMO,  
CONCAT(((MAX(pr.buyPrice)-MIN(pr.buyPrice))*1000),'$') AS DIFERENCIA FROM products pr;
```

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
1 • SELECT MAX(pr.buyPrice) AS MAXIMO, MIN(pr.buyPrice) AS MINIMO,  
2   CONCAT(((MAX(pr.buyPrice)-MIN(pr.buyPrice))*1000),'$') AS DIFERENCIA  
3   FROM products pr;
```

Below the query editor, the 'Result Grid' is displayed with the following data:

	MAXIMO	MINIMO	DIFERENCIA
▶	103.42	15.91	87510.00\$

**11-Muestra cuantos empleados tiene cada jefe(lastName) a su cargo. Usa el campo reportsTo.**

```
SELECT employeeNumber, firstName, lastName, COUNT(reportsTo) FROM employees GROUP  
BY reportsTo;
```

The screenshot shows a SQL IDE interface. The query editor contains the following SQL code:

```
1 SELECT employeeNumber, firstName, lastName, COUNT(reportsTo) FROM employees GROUP BY reportsTo;
```

Below the query editor, the 'Result Grid' is displayed with the following data:

	employeeNumber	firstName	lastName	COUNT(reportsTo)
▶	1002	Diane	Murphy	0
	1056	Mary	Patterson	2
	1088	William	Patterson	4
	1611	Andy	Fixter	3
	1337	Loui	Bondur	6
	1165	Leslie	Jennings	6
	1625	Yoshimi	Kato	1

**12-Añade el campo birthdate (fecha de nacimiento) a la tabla de empleados (employees) y haz que todos os empleados tengan el 01/01/2000 en dicho campo. Puedes usar varias sentencias sql si lo ves necesario.**

ALTER TABLE employees ADD COLUMN birthdate date DEFAULT '2000-01-01';

