La sentencia EXPLAIN

MySQL nos ofrece también facilidades a la hora de evaluar las sentencias SQL, gracias a la sentencia EXPLAIN

Presentamos primero la ejecución de una sentencia SQL más o menos compleja:

```
mysql> select productos.clave, concat(parte,' ',tipo,' ', especificación) as producto, proveedores.em-
presa , precio , pago from productos natural join precios natural join proveedores;
+-------+---------------------------+----------+--------+------------------+
| clave | producto                  | empresa  | precio | pago             |
+-------+---------------------------+----------+--------+------------------+
|  003  | Procesador 1.7 GHz 64 bits | Tecno-k  | 198.34 | crédito          |
|  005  | RAM 128MB 333 MHz         | Tecno-k  | 9.27   | crédito          |
|  006  | RAM 256MB 400 MHz         | Tecno-k  | 34.85  | crédito          |
|  007  | Disco Duro 80 GB 7200 rpm | Tecno-k  | 59.95  | crédito          |
|  010  | Disco Duro 40 GB 4200 rpm | Tecno-k  | 61.22  | crédito          |
|  012  | Monitor 1024x876 60 Hz    | Tecno-k  | 62.29  | crédito          |
|  001  | Procesador 2 GHz 32 bits  | Patito   | 30.40  | efectivo         |
|  002  | Procesador 2.4 GHz 32 bits | Patito   | 33.63  | efectivo         |
|  003  | Procesador 1.7 GHz 64 bits | Patito   | 195.59 | efectivo         |
|  005  | RAM 128MB 333 MHz         | Patito   | 9.78   | efectivo         |
|  006  | RAM 256MB 400 MHz         | Patito   | 32.44  | efectivo         |
|  007  | Disco Duro 80 GB 7200 rpm | Patito   | 59.99  | efectivo         |
|  010  | Disco Duro 40 GB 4200 rpm | Patito   | 62.02  | efectivo         |
|  001  | Procesador 2 GHz 32 bits  | Nacional | 30.82  | crédito,efectivo |
|  002  | Procesador 2.4 GHz 32 bits | Nacional | 32.73  | crédito,efectivo |
|  003  | Procesador 1.7 GHz 64 bits | Nacional | 202.25 | crédito,efectivo |
|  005  | RAM 128MB 333 MHz         | Nacional | 9.76   | crédito,efectivo |
|  006  | RAM 256MB 400 MHz         | Nacional | 31.52  | crédito,efectivo |
|  007  | Disco Duro 80 GB 7200 rpm | Nacional | 58.41  | crédito,efectivo |
|  010  | Disco Duro 40 GB 4200 rpm | Nacional | 64.38  | crédito,efectivo |
+-------+---------------------------+----------+--------+------------------+
20 rows in set (0.00 sec)
```

Ahora utilizamos la sentencia EXPLAIN para que MySQL nos explique cómo ha realizado esta consulta:

mysql> explain select productos.clave, concat(parte,' ',tipo,' ', especificación) as producto, proveedores.empresa , precio , pago from productos natural join precios natural join proveedores;

```
+-------------+--------+---------------+---------+---------+---------------+------+------------+
| table       | type   | possible_keys | key     | key_len | ref           | rows | Extra      |
+-------------+--------+---------------+---------+---------+---------------+------+------------+
| precios     | ALL    | NULL          | NULL    |  NULL   | NULL          | 20   |            |
| productos   | eq_ref | PRIMARY       | PRIMARY |    4    | precios.clave|  1   |            |
| proveedores | ALL    | PRIMARY       | NULL    |  NULL   | NULL          | 3    | where used |
+-------------+--------+---------------+---------+---------+---------------+------+------------+
3 rows in set (0.00 sec)
```

En cada fila del resultado, nos explica cómo ha utilizado los índices de cada tabla involucrada en la consulta. La columna 'type' nos indica el tipo de "join" que ha podido hacer. En nuestro caso, 'eq_ref', 'ref' o 'ref_or_null' indica que se ha consultado una fila de esta tabla para cada combinación de filas de las otras. Es una buena señal, se están utilizando los índices, tal como indican el resto de columnas (en concreto el atributo 'clave' que es su clave primaria).

Vemos que en las otras dos tablas, el tipo de 'join' es ALL, esto indica que el gestor ha tenido que leer toda la tabla para comprobar las condiciones que le hemos exigido en la consulta. En el caso de la tabla proveedores, habría podido utilizar la clave primaria ('possible_keys'), pero no lo ha hecho.

Vamos a intentar mejorar esta consulta. Vemos que en la tabla precios no se ha definido ningún índice, lo que facilitaría la labor al SGBD:

```
mysql> alter table precios add index empresa_idx (empresa);
Query OK, 20 rows affected (0.00 sec)
Records: 20 Duplicates: 0 Warnings: 0

mysql> alter table precios add index clave_idx (clave);
Query OK, 20 rows affected (0.00 sec)
Records: 20 Duplicates: 0 Warnings: 0

mysql> explain select productos.clave, concat(parte,' ',tipo,' ', especificación) as producto,
proveedores.empresa , precio , pago from productos natural join precios natural join proveedores;
+------------+-------+---------------------+------------+---------+-------------+-----+-----------+
| table      | type  | possible_keys       | key        | key_len | ref         |rows | Extra     |
+------------+-------+---------------------+------------+---------+-------------+-----+-----------+
| proveedores | ALL   | PRIMARY             | NULL       | NULL    | NULL        | 3   |           |
| precios    | ref   | empresa_idx,clave_idex| empresa_idx | 20      | productos.emp| 7   |           |
| productos  | eq_ref | PRIMARY             | PRIMARY    | 4       | precios.clave| 1   |           |
+------------+-------+---------------------+------------+---------+-------------+-----+-----------+
3 rows in set (0.00 sec)
```

Las cosas han cambiado sustancialmente. El gestor ha pasado de leer 24 filas de datos, a leer 11. También ha cambiado el orden de lectura de las tablas, haciendo primero una lectura total de la tabla proveedores (que es inevitable ya que no hemos puesto ninguna condición en el SELECT) y, después, ha aprovechado los índices definidos en 'precios' y en 'productos'.