

**Desarrollo de Aplicaciones Multiplataforma****Grupo DAM 1º**

(BBDD\_pedidos1.sql)

**Ejercicio Repaso 1**

Una vez cargada la base de datos si hacéis el siguiente `SELECT` sobre la tabla `productos` vais a ver una salida similar a esta (solo muestro los primeros diez registros):

```
mysql> select IdProducto, UnidadesEnPedido from productos limit 10;
```

idproducto	UnidadesEnPedido
1	NULL
2	NULL
3	NULL
4	NULL
5	NULL
6	NULL
7	NULL
8	NULL
9	NULL
10	NULL

```
10 rows in set (0.00 sec)
```

Se pide crear un procedimiento almacenado que se encargue de actualizar en la tabla `productos` la columna `UnidadesEnPedido` que esta toda a `NULL`. Para ello el procedimiento deberá ir revisando y acumulando todas las cantidades pedidas de cada producto que aparecen reflejadas en la tabla `detalles_pedido`.

El procedimiento se puede realizar de varias formas. Se valorara aquel procedimiento que minimice el numero de actualizaciones (`updates`) a realizar teniendo en cuenta que :

```
mysql> select count(*) from productos;
```

count(*)
77

```
1 row in set (0.00 sec)
```

```
mysql> select count(*) from detalles_pedido;
```

count(*)
2155

```
1 row in set (0.00 sec)
```

Después de lanzar el procedimiento si volvéis a lanzar la `select` sobre la tabla `productos` deberíais tener una salida similar a esta (limitada a diez registros):

```
mysql> select IdProducto, UnidadesEnPedido from productos limit 10;
```

idproducto	unidadesenpedido
1	828
2	1057
3	328
4	453
5	298
6	301
7	763
8	372
9	95
10	742

```
10 rows in set (0.00 sec)
```

**Ejercicio Repaso 2**

Se pide hacer un disparador que actualice las columnas `UnidadesEnPedido` y `UnidadesEnExistencia` de la tabla `productos` cada vez que se inserte una fila en la tabla `detalles_de_pedido` siempre que haya suficientes unidades para dar de alta esa línea de pedido. Si no es así el alta no se produce avisando al usuario del motivo.