

1)Realiza una transacción que elimine todas los empleados que no tienen un departamento asociado (BBDD Empleados) .

```
START TRANSACTION;  
DELETE FROM empleado WHERE codigo_departamento IS NULL;  
COMMIT;
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0002 segundos.)

```
START TRANSACTION;
```

[Editar en línea] [Editar] [Crear código PHP]

✓ 2 filas eliminadas. (La consulta tardó 0,0004 segundos.)

```
DELETE FROM empleado WHERE codigo_departamento IS NULL;
```

[Editar en línea] [Editar] [Crear código PHP]

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0012 segundos.)

```
COMMIT;
```

[Editar en línea] [Editar] [Crear código PHP]

2.-Modifica la tabla detalle_pedido para insertar un campo numérico llamado iva. Mediante una transacción, establece el valor de ese campo a 18 para aquellos registros cuyo pedido tenga fecha a partir de Enero de 2009. A continuación actualiza el resto de pedidos estableciendo el iva al 21. (BBDD Jardinería)

```
ALTER TABLE detalle_pedido ADD iva NUMERIC(5,2);
```

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0004 segundos.)

```
ALTER TABLE detalle_pedido ADD iva NUMERIC(5,2);
```

[Editar en línea] [Editar] [Crear código PHP]

```
START TRANSACTION;
UPDATE detalle_pedido SET iva=18
WHERE fecha_pedido >= '2009-01-01';
```

```
UPDATE detalle_pedido SET iva=21
WHERE fecha_pedido < '2009-01-01';
```

```
COMMIT;
```

Podemos ver que la tabla de detalle de pedido no tiene fecha_pedido por eso nos da error a la hora de ejecutarlo.

La columna 'fecha_pedido' en where clause es desconocida

3.-Realiza una transacción que inserte un pedido para cada uno de los clientes que creaste en el ejercicio 14 del tema anterior (ejercicio 3 Insert, Update y Delete). Cada pedido debe incluir dos productos. (BBDD Jardinería)

4.-Realiza una transacción que realice los pagos de los pedidos que han realizado los clientes del ejercicio anterior. (BBDD Jardinería)

```
START TRANSACTION;
```

```
SELECT codigo_cliente, SUM(precio_unidad* cantidad) AS total_pagar
FROM detalle_pedido
INNER JOIN producto ON detalle_pedido.codigo_producto= producto.codigo_producto
INNER JOIN pedido ON detalle_pedido.codigo_pedido = pedido.codigo_pedido
GROUP BY codigo_cliente;
```

```
UPDATE pedido SET estado = 'pagado';
```

```
COMMIT;
```

codigo_cliente	total_pagar
1	10365.00
3	13726.00
4	81849.00
5	23794.00
7	2390.00
9	929.00
13	2246.00
14	4160.00
15	12081.00
16	7899.00
19	232.00
23	272.00
26	18979.00
27	10972.00
28	8489.00
30	11363.00
35	3321.00
36	3500.00
38	1171.00

5.-Considera que tenemos una tabla donde almacenamos información sobre cuentas bancarias definida de la siguiente manera:

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

```
START TRANSACTION;
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

✓ 0 filas afectadas. (La consulta tardó 0,0006 segundos.)

```
UPDATE cuentas SET saldo = saldo - 100 WHERE id = 20;
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

✓ 0 filas afectadas. (La consulta tardó 0,0003 segundos.)

```
UPDATE cuentas SET saldo = saldo + 100 WHERE id = 30;
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0002 segundos.)

```
COMMIT;
```

[\[Editar en línea \]](#) [\[Editar \]](#) [\[Crear código PHP \]](#)

¿Qué ocurriría si el sistema falla o si se pierde la conexión entre el cliente y el servidor después de realizar la primera sentencia UPDATE?

Podemos ver que la transacción se parará y no se completará del todo. En nuestro caso podemos ver que la cuenta nº20 no se actualizará con la de 100 euros que se pretendía realizar en dicha transacción. Esto a día de hoy podría provocar problemas de contabilidad en cualquier banco.

¿Qué ocurriría si no existiese alguna de las dos cuentas (id = 20 y id = 30)?

Pues muy sencillo, esta transacción no se completaría ya que con la sentencia UPDATE no se actualizará ningún saldo.

¿Qué ocurriría en el caso de que la primera sentencia UPDATE falle porque hay menos de 100 € en la cuenta y no se cumpla la restricción del CHECK establecida en la tabla?

No se realizará ninguna actualización de saldo en ninguna de las cuentas y los saldos no sufrirán ningún cambios como estaban antes de la transacción. Esto se debe al 'CHECK' lo que hace que el saldo de la cuenta sea siempre positivo.