**Transacciones**

Las transacciones de bases de datos permiten agrupar sentencias en bloques, que van a ser ejecutados simultáneamente de tal forma que podamos evaluar si alguna de las sentencias ha fallado y de ser así poder deshacer los cambios en el momento sin alterar de forma alguna la base de datos.

Si necesitas utilizar transacciones deberás asegurarte de que estén soportadas por el motor de almacenamiento que gestiona tus tablas en MySQL. Si utilizas InnoDB, por defecto cada consulta individual se incluye dentro de su propia transacción. Puedes gestionar este comportamiento con el método **autocommit** (función **mysqli\_autocommit**).

$conexion->autocommit(false); // deshabilitamos el modo transaccional automático

Al deshabilitar las transacciones automáticas, las siguientes operaciones sobre la base de datos iniciarán una transacción que deberás finalizar utilizando:

* **commit** (o la función **mysqli\_commit**). Realizar una operación "commit" de la transacción actual, devolviendo true si se ha realizado correctamente o false en caso contrario.
* **rollback** (o la función **mysqli\_rollback**). Realizar una operación "rollback" de la transacción actual, devolviendo true si se ha realizado correctamente o false en caso contrario.

Ejemplo:

Dar de alta: insert into producto values('produc1',NULL,'prd1',NULL,0,'MP3')

Y dar de alta tbien: insert into stock values('produc1',1,60)

Al escribir este último insert nos equivocamos y en vez de produc1 ponemos produc11.

Sin transacciones daría de alta en t.producto y error en la insert a la t.stock

Ejercicio: Realizar el ejercicio anterior y comprobar que se produce el alta en productos y no en stock.

El ejercicio consiste en dar de alta en las 2 tablas, o si se produce algún error en alguna de las 2 insert no se ejecutará ninguna.

<?php

@$conexion = **new** mysqli('localhost', 'root', '', 'dwes');

$error = $conexion->connect\_errno;

**if** ($error == **null**) {

$conexion->autocommit(**false**);

$resultado1 = $conexion->query("insert into producto values('produc1',NULL,'prd1',NULL,0,'MP3')");

**if** ($resultado1) {

**print** "<p>Se ha insertado **$conexion**->affected\_rows registro.</p>";

}

**else** **echo** "error1";

$resultado2 = $conexion->query("insert into stock values('produc11',1,60)"); //error al escribir, se queria poner: produc1

**if** ($resultado2) {

**print** "<p>Se ha insertado **$conexion**->affected\_rows registro.</p>";

}

**else** **echo** "error2";

**if**(!$resultado1 && !$resultado2)$conexion->commit();**else** $conexion->rollback();

}

$conexion->close();

?>

**Ejercicio: Mejorar el ejercicio anterior : print** "<p>Se ha insertado **$conexion**->affected\_rows registro.</p>"; aparece aunque no se haya dado de alta

Ejercicio :

 La tabla stock de la base de datos dwes, la tienda 1 (CENTRAL) tiene 2 unidades del producto de código 3DSNG y la tienda 3 (SUCURSAL2) ninguno. Suponiendo que los datos son esos (no hace falta que los compruebes en el código), utiliza una transacción para mover una unidad de ese producto de la tienda 1 a la tienda 3.