

PRACTICA 2

Realiza las siguientes consultas en la **base de datos ebanca**:

1. Inicia una transacción en la que se va a transferir 100 € de la cuenta 3 a la cuenta 4.

Generar los registros correspondientes en la tabla movimientos.

```
Set autocommit = 0;
Start transaction;
Update cuentas set saldo=saldo-100 where cod_cuenta="3";
Update cuentas set saldo=saldo+100 where cod_cuenta="4";
Insert into movimientos values(125, now(), -100, 33333333,3);
Insert into movimientos values(126, now(), 100, 44444444,4);
Commit work;
```

2. ¿Qué ocurre si en mitad de una transacción sin confirmar modificamos la variable autocommit para ponerla a 1? ¿Se abortará o confirmará la transacción o habrá un resultado imprevisto?

Si durante la transacción modificamos la variable autocommit a 1, la transacción se verá finalizada ya que 1 es = false;

3. El cliente 3 abre una nueva cuenta con la fecha del día de hoy y saldo 1000€.

```
Set autocommit = 0;
Start transaction;
Insert into cuentas values(now(), 1000, 36,3);
Insert into movimientos values(127, now(), 1000, 33333333,3);
Commit work;
```

4. Incrementa en un 10% el saldo del cliente con más dinero.

```
Select cod_cliente into @clientemax from cuentas group by
cod_cliente order by sum(saldo) desc limit 1;
Set autocommit = 0;
Start transaction;
Update cuentas set saldo = saldo+saldo*10/100 where
cod_cliente=@clientemax;
Commit work;
```

Unidad 5 - Edición de datos

5. Agrega un campo en la columna clientes llamada VIP de tipo booleano. A continuación, rellena esta columna con Verdadero para los clientes que tienen un saldo superior a 50.000€ y falso en caso contrario.

```
Alter table clientes add VIP Boolean;
Set autocommit = 0;
Start transaction;
Update clientes set VIP= 0;
Update clientes set VIP=1 where codigo_cliente in (Select
cod_cliente from cuentas group by cod_cliente having
sum(saldo)>50000 order by sum(saldo));
Update clientes set VIP=0 where codigo_cliente in (Select
cod_cliente from cuentas group by cod_cliente having
sum(saldo)<=50000 order by sum(saldo));
Commit work;
```

6. El cliente numero 2 quiere reunir todo su dinero en una única cuenta.

```
Set autocommit = 0;
Start transaction;
select sum(saldo) into @saldototal from cuentas where
cod_cliente=2 group by cod_cliente;
delete from movimientos where dni=22222222;
Insert into cuentas values(now(), @saldototal, 37,2);
Insert into movimientos values(128, now(), @saldototal,
22222222,2);
delete from cuentas where cod_cliente=2 and cod_cuenta!=37;
Commit work;
```

7. Elimina las cuentas con saldo cero y cuyo cliente no haya realizado ningún movimiento en el último año.

```
Set autocommit = 0;
Start transaction;
select distinct c.cod_cuenta into @codcuentasdelete from
cuentas c, movimientos m where c.saldo=0 and
c.cod_cuenta=m.cod_cuenta and year(fechahora)<2014;
Delete from movimientos where cod_cuenta=@codcuentasdelete;
```

Unidad 5 - Edición de datos

```
Delete from cuentas where cod_cuenta=@codcuentasdelete;  
Commit work;
```

@idanirf