

# SQL COMO DML: MODIFICACIÓN DE DATOS

---

Prácticas

Manuel Ramos Cabrer

# Modificación de Datos - Inserción

- Añadir una nueva tupla a *cuenta*

```
insert into cuenta  
values ( 'A-9732' , 'Vigo' ,1200)
```

o, como forma alternativa

```
insert into cuenta (nombre-sucursal, saldo, numero-  
cuenta)  
values ( 'Vigo' , 1200, 'A-9732' )
```

- Añadir una nueva tupla a *cuenta* con *saldo* puesto a nulo

```
insert into cuenta  
values ( 'A-777' , 'Vigo' , null)
```

# Modificación de datos – Inserción

- Dar como premio a todos los clientes con préstamo en la sucursal de Vigo una nueva cuenta de ahorro con 200 euros de saldo. El número de préstamo servirá como número de cuenta para la nueva cuenta de ahorro.

**insert into** *cuenta*

**select** *numero-prestamo, nombre-sucursal, 200*

**from** *prestamo*

**where** *nombre-sucursal = 'Vigo'*

**insert into** *depositante*

**select** *nombre-cliente, numero-prestamo*

**from** *prestamo, prestatario*

**where** *nombre-sucursal = 'Vigo'*

**and** *prestamo.numero-cuenta =*

*prestatario.numero-cuenta*

- La sentencia select-from-where se evalúa totalmente antes de insertar ninguno de sus resultados en la relación (si no, consultas como

**insert into** *tabla1* **select \* from** *tabla1*

causarían problemas

# Modificación de datos – Actualizaciones

- Incrementar todas las cuentas con más de 10,000 euros un 6% y el resto de cuentas un 5%.

- Escribimos dos sentencias **update**:

```
update cuentas  
set saldo = saldo * 1.06  
where saldo > 10000
```

```
update cuentas  
set saldo = saldo * 1.05  
where saldo <= 10000
```

- El orden es importante
- Se puede hacer mejor con la sentencia **case**

## Sentencia Case para actualizaciones condicionales

- La misma consulta de antes: aumentar los saldos de todas las cuentas de más de 10,000 euros un 6% y las demás un 5%.

**update** *cuenta*

**set** *saldo* = **case**

**when** *saldo* <= 10000 **then** *saldo* \* 1.05

**else** *saldo* \* 1.06

**end**

# Modificación de datos – Borrado

- Borrar todas las cuentas de la sucursal de Vigo

```
delete from cuenta  
where nombre-sucursal = 'Vigo'
```

- Borrar todas las cuentas de todas las sucursales de Madrid.

```
delete from cuenta  
where nombre-sucursal in (select nombre-sucursal  
                                from sucursal  
                                where ciudad-sucursal =  
                                'Madrid' )  
delete from depositante  
where numero-cuenta in  
      (select numero-cuenta  
            from sucursal, cuenta  
            where ciudad-sucursal = 'Madrid'  
            and sucursal.nombre-sucursal =  
                cuenta.nombre-sucursal)
```

# Ejemplo de borrado

- Borrar todas las cuentas con saldos por debajo de la media del banco.

```
delete from cuenta  
  where saldo < (select avg (saldo)  
    from cuenta)
```

- ★ Problema: a medida que borramos tuplas, la media cambia
- ★ Solución utilizada en SQL:
  1. Primero, calcular **avg** y encontrar todas las tuplas a borrar
  2. Segundo, borrar todas las tuplas encontradas antes (sin recalcular **avg** ni recomprobar las tuplas)