

# Diseño de Bases de Datos



# SQL

## DML

Sucursal = (nombreSucursal, ciudadSucursal, activo)

Cliente = (codCliente, dni, nombCliente, dirCliente, viveCliente)

Prestamo = (nroPrestamo, importe, nombreSucursal(fk))

PropietarioPrestamo = (codCliente(fk), nroPrestamo(fk))

Cuenta = (nroCuenta, saldo, nombreSurcursal(fk))

PropietarioCuenta = (nroCuenta(fk), codCliente(fk))

- Ej: clientes con cuentas o préstamos en cualquier sucursal
- Ej: clientes con cuentas y préstamos en cualquier sucursal
- Ej: clientes con cuentas y sin préstamos

# SQL

## DML

### 1. Clientes con cuentas o préstamos en cualquier sucursal

```
(Select dni, nombCliente  
From PropietarioCuenta natural join cliente) Union  
(Select dni, nombCliente  
From PropietarioPrestamo natural join cliente)
```

### 2. Clientes con cuentas y préstamos en cualquier sucursal

```
(Select dni, nombCliente  
From PropietarioCuenta natural join cliente) Intersect  
(Select dni, nombCliente From PropietarioPrestamo natural  
join cliente)
```

### 3. Clientes con cuentas y sin préstamos

```
(Select dni, nombCliente  
From PropietarioCuenta natural join cliente) Except  
(Select dni, nombCliente  
From PropietarioPrestamo natural join cliente)
```

# SQL

## DML

Sucursal = (nombreSucursal, ciudadSucursal, activo)

Cliente = (codCliente, dni, nombCliente, dirCliente, viveCliente)

Prestamo = (nroPrestamo, importe, nombreSucursal(fk))

PropietarioPrestamo = (codCliente(fk), nroPrestamo(fk))

Cuenta = (nroCuenta, saldo, nombreSurcursal(fk))

PropietarioCuenta = (nroCuenta(fk), codCliente(fk))

- Ej: cantidad de cuentas con saldo mayor a \$50000
- Ej: saldo promedio de las cuentas de la sucursal 'X'
- Ej: importe del mayor préstamo otorgado por la sucursal 'Y'
- Ej: importe total asignado a prestamos

**Cantidad de cuentas con saldo mayor a \$50000**

**Select** COUNT(nroCuenta) as cantCuentas    ☐ podria contar cualquier campo

**From** Cuenta

**Where** saldo > 50000

**Saldo promedio de las cuentas de la sucursal 'X'**

**Select** AVG(saldo) as Promedio

**From** Cuenta

**Where** nombreSucursal = 'X'

**Importe del mayor préstamo otorgado por la sucursal 'Y'**

```
Select MAX(importe) as Maximo  
From Prestamo  
Where nombreSucursal = 'Y'
```

**Importe total asignado a prestamos**

```
Select Sum(importe) as importe  
From Prestamo
```

# SQL

## DML

Sucursal = (nombreSucursal, ciudadSucursal, activo)

Cliente = (codCliente, dni, nombCliente, dirCliente, viveCliente)

Prestamo = (nroPrestamo, importe, nombreSucursal(fk))

PropietarioPrestamo = (codCliente(fk), nroPrestamo(fk))

Cuenta = (nroCuenta, saldo, nombreSurcursal(fk))

PropietarioCuenta = (nroCuenta(fk), codCliente(fk))

- Ej: obtener saldo promedio de las cuentas de c/ sucursal
- Ej: presentar las sucursales junto con el saldo promedio de sus cuentas siempre y cuando supere los \$200.000
- Ej: contar el N° de clientes que tienen cuentas en cada sucursal
- Ej: saldo promedio de las cuentas de c/ cliente que vive en La Plata y tiene al menos 3 cuentas

**Obtener saldo promedio de las cuentas de c/ sucursal**

```
Select nombreSucursal, avg(saldo) as saldoProm  
From cuenta  
Group by nombreSucursal
```

**Presentar las sucursales y su saldo promedio siempre y cuando supere los \$200.000**

```
Select nombreSucursal, avg(saldo)  
From cuenta  
Group by nombreSucursal  
Having avg(saldo) > 200000
```



**Contar el N° de clientes que tienen cuentas de cada sucursal**

```
Select nombreSucursal, count(distinct codCliente)  
From cuenta inner join propietarioCuenta  
on( cuenta.nroCuenta = propietarioCuenta.nroCuenta)  
Group by nombreSucursal
```

**Saldo promedio de las cuentas de c/ cliente que vive en La Plata y tiene al menos 3 cuentas**

```
Select nombCliente, avg(saldo)  
From propietarioCuenta inner join cuenta  
(propietarioCuenta.nroCuenta= cuenta.nroCuenta )  
Inner join cliente on (propietarioCuenta. codcliente =  
cliente.codCliente )  
Where viveCliente = "La Plata"  
Group by propietarioCuenta.codcliente, cliente.nombCliente  
Having count (propietarioCuenta.nroCuenta) >= 3
```

**Cientes con préstamos y cuentas en cualquier sucursal (alternativa a  $\cap$ )**

**Select distinct** nombCliente

**From** propietarioprestamo **inner join** cliente on  
(propietarioprestamo. codcliente = cliente.codcliente )

**Where** codcliente **in** (

**Select** codcliente

**From** propietariocuenta

)

**Cientes con préstamos y cuentas en una misma sucursal de “La Plata”**

**Select distinct** nombCliente

**From** propietarioprestamo pp **inner join** prestamo p  
on(pp.nroPréstamo = p.nroPrestamo) **inner join** sucursal s on  
(p.nombreSucursal = s.nombreSucursal)

**inner join** cliente c on(pp.codCliente=c.codCliente )

**Where** s.ciudadSucursal=“La Plata” **and** c.codCliente in

(**Select** codCliente

**From** propietariocuenta pc **inner join** cuenta c on (pc.nroCuenta =  
c.nroCuenta )

**Where** c.nombreSucursal = **s.nombreSucursal**)-> s corresponde a  
la consulta ppal

# SQL

## DML

Sucursal = (nombreSucursal, ciudadSucursal, activo)

Cliente = (codCliente, dni, nombCliente, dirCliente, viveCliente)

Prestamo = (nroPrestamo, importe, nombreSucursal(fk))

PropietarioPrestamo = (codCliente(fk), nroPrestamo(fk))

Cuenta = (nroCuenta, saldo, nombreSurcursal(fk))

PropietarioCuenta = (nroCuenta(fk), codCliente(fk))

- Ej: presentar las sucursales que tengan activo mayor que alguna otra
- Ej: presentar la sucursal que tenga activo superior a todas
- Ej: encontrar la sucursal que tiene el mayor saldo promedio
- Ej: clientes con cuenta en todas las sucursales

Presentar las sucursales que tengan activo mayor que alguna otra

```
Select nombreSucursal  
From sucursal  
Where activo > some ( select activo  
                        from sucursal)
```

Presentar la sucursal que tenga activo superior a todas

```
Select nombreSucursal  
From sucursal  
Where activo > =all ( select activo  
                      from sucursal)
```

Encontrar la sucursal que tiene el mayor saldo promedio

```
Select nombreSucursal  
From cuenta  
Group by nombreSucursal  
Having avg (saldo) >= all (  
    Select avg (saldo)  
    From cuenta  
    Group by nombreSucursal  
)
```

**Cliente con cuentas en todas las sucursales**

```
Select nombCliente  
From cliente  
where not exists  
(select * from sucursal s  
  Where not exists(select * from propietarioocuenta pc  
    inner join cuenta c on (pc.nroCuenta=c.nroCuenta)  
    where c.nombreSucursal=s.nombreSucursal and  
    pc.codCliente=cliente.codCliente))-> corresponden a las  
consultas de nivel superior
```