

SQL - DDL

- CREATE DATABASE nombreBD → crea BD
- DROP DATABASE nombreBD → borra contenidos y estructura BD
- ALTER DATABASE nombreBD (ALTER/MODIFY NAME = nuevoNombreBD)
- CREATE TABLE empresa (
idEmpresa INTEGER NOT NULL AUTO_INCREMENT,
empresa VARCHAR(100) NOT NULL,
...,
PRIMARY KEY(idEmpresa)
);
- CREATE TABLE empleado (
idEmpleado INTEGER NOT NULL AUTO_INCREMENT,
idEmpresa INTEGER UNSIGNED NOT NULL,
nombreEmpleado VARCHAR(100) NOT NULL,
...,
PRIMARY KEY(idEmpleado),
FOREIGN KEY(idEmpresa)
REFERENCES empresas(idEmpresa)
ON DELETE Restrict
ON UPDATE No Action
);
- DROP TABLE nombreTabla
- ALTER TABLE empresa (
ADD column razonSocial VARCHAR(100) NOT NULL,
DROP column direccion
ALTER/MODIFY column empresa VARCHAR(120) NOT NULL
);
- CREATE [UNIQUE] INDEX nombreIndice ON tabla [USING nombreAcceso] (columna [ASC | DESC])
 - UNIQUE: el sistema comprueba si existen valores duplicados en la tabla cuando se crea el índice (si ya existen datos) y cada vez que se añaden datos. Los intentos de insertar o actualizar datos duplicados generarán un error.
 - USING nombreAcceso (BTREE, HASH, etc)
- DROP INDEX nombreIndice

SQL - DML

Sucursal = (idSuc, nombreSuc, ciudadSuc, activo)

Cliente = (idCte, nombreCte, direccionCte, ciudadCte)

Prestamo = (idSuc, nroPrestamo, importe)

PropietarioPrestamo = (idCte, nroPrestamo)

Cuenta = (idSuc, nroCuenta, saldo)

PropietarioCuenta = (idCte, nroCuenta)

Ej 1)

Select *

From sucursal

Ej 2)

Select distinct idSuc

From prestamo

Ej 3)

Select nroPrestamo, importe * 1.15 [**AS importeEstimado**]

From préstamo

Ej 4)

Select nroPrestamo

From Prestamo

Where idSuc = 37 and importe > 200000

Ej 5)

Select nroPrestamo

From Prestamo

Where importe between 20000 and 100000

Ej 6)

Select nombreCte, PP.nroPrestamo

From Cliente as C, Prestamo as P, PropietarioPrestamo as PP

Where C.idCte = PP.idCte

and PP.nroPrestamo = P. nroPrestamo

and idSuc = 64

Ej 7)

Select distinct T.idSuc, T.nombreSuc

From sucursal as T, sucursal as S

Where T.activo > S.activo and S.ciudadSuc = "Buenos Aires"

Ej 8)

Select idCte, nombreCte

From Cliente

Where nombreCte Like "G%"

and direccionCte Like "%Los Hornos%"

Ej 9)

Select nombreCte

From cliente

Order by nombreCte

Ej 10)

Select nroPrestamo, nombreSuc, importe

From Prestamo P, Sucursal S

Where P.idSuc = S.idSuc

Order by nombreSuc [ASC], importe DESC

Ej 11)

(Select idCte From PropietarioCuenta)

Union

(Select idCte From PropietarioPrestamo)

Ej 12)

(Select idCte From PropietarioCuenta)

Intersect

(Select idCte From PropietarioPrestamo)

Ej 13)

(Select idCte From PropietarioCuenta)

Except

(Select idCte From PropietarioPrestamo)

Ej 14)

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

From Cuenta

Where saldo > 50000

Ej 15)

Select AVG(saldo) as Promedio

From Cuenta C, Sucursal S

Where C.idSuc = S.idSuc

and nombreSuc = 'LP-1'

Ej 16)

Select MAX(importe) as Maximo

From Prestamo P, Sucursal S

Where P.idSuc = S.idSuc

and nombreSuc = 'LP-2'

Ej 17)

Select SUM(importe) as total

From Prestamo

Ej 18)

Select idSuc, avg(saldo) as saldoProm

From cuenta

Group by idSuc

Ej 19) → un cliente solo debe contarse una vez aunque tenga varias cuentas

Select idSuc, count(distinct idCte)

From cuenta C, propietarioCuenta PC

Where C.nroCuenta = PC. nroCuenta

Group by idSuc

Ej 20)

Select idSuc, avg(saldo) as saldoPromedio

From cuenta

Group by idSuc

Having avg(saldo) > 200000

Ej 21)

Select CL.idCte, nombreCte, avg(saldo)

From cuenta C, propietarioCuenta PC, cliente CL

Where C.nroCuenta = PC.nroCuenta

and PC.idCte = CL.idCte

and ciudadCte = "La Plata"

Group by CL.idCte, nombreCte → se debe incluir 'nombreCte' porque aparece en el SELECT

Having count (CL.nroCuenta)>=3

Ej 22)

Select nroPrestamo From prestamo

Where importe is null

Ej 23)

Select distinct idCte

From propietarioPrestamo

Where idCte in (Select idCte From propietarioCuenta)

Ej 24)

Select distinct idCte

From propietarioprestamo as PP, prestamo as P, sucursal as S

Where PP.nroPrestamo = P.nroPrestamo

and P.idSuc = S.idSuc and S.ciudadSuc= "La Plata"

and PP.idCte in (Select idCte

From propietariocuenta as PC, cuenta as C, sucursal as S1

Where PC.nroCuenta = C.nroCuenta

and C.idSuc = S1.idSuc

and S1.ciudadSuc= "La Plata")

Y SI EL PRESTAMO Y LA CUENTA DEBEN SER EN LA MISMA SUCURSAL DE LA PLATA?

Select distinct idCte

From propietarioprestamo as PP, prestamo as P, sucursal as S

Where PP.nroPrestamo = P.nroPrestamo

and P.idSuc = S.idSuc and S.ciudadSuc="La Plata"

and PP.idCte in (Select idCte

From propietariocuenta as PC, cuenta as C

Where PC.nroCuenta = C.nroCuenta

and C.idSuc = S.idSuc) → **USAMOS LA MISMA TABLA**

Ej 25)

Select idSuc, nombreSuc

From sucursal

Where activo > some (select activo from sucursal)

Ej 26)

Select nombreSuc

From sucursal

Where activo >= all (select activo from sucursal) → **mayor o igual, por si hay más de un máximo**

Ej 27)

```
Select S.idSuc, nombreSuc
From cuenta as C, sucursal as S
Where C.idSuc = S.idSuc
Group by S.idSuc
Having avg (saldo) >= all ( select avg saldo
                           from cuenta
                           group by idSuc)
```

Ej 28)

```
Select idCte
From propietarioprestamo PP
Where Exist
      ( Select *
        From propietariocuenta as PC
        Where PP.idCte = PC.idCte)
```

Ej 29)

```
Select distinct idCte
From PropietarioCuenta as PC
Where Not Exist (      (Select idSuc
                        From sucursal
                        Where ciudadSuc = "Buenos Aires") → suc. BsAs
                  Except
                  ( Select C.idSuc
                    From cuenta as C
                    Where C.nroCuenta= PC.nroCuenta) → suc. propietario analizado
                )
```

Ej 30)

Select PC.idCte

From PropietarioCuenta **as** PC

Where unique (

Select idCte

From Cuenta **as** C, Sucursal **as** S, PropietarioCuenta **as** PC2

Where C.nroCuenta = PC.nroCuenta

And PC.nroCuenta = PC2.nroCuenta

And C.idSuc = S.idSuc

And S.nombreSuc = "SL1")

Ej 31)

CREATE VIEW clientesActivos **AS**

(**SELECT** S.idSuc, nombreSuc, PP.idCte, nombreCte, direccionCte

FROM Sucursal **as** S, Prestamo **as** P, PropietarioPrestamo **as** PP, Cliente **as** CL

WHERE S.idSuc = P.idSuc

AND P.nroPrestamo = PP.nroPrestamo

AND PP.idCte = CL.idCte

AND P.importe IS NOT NULL

UNION

(**SELECT** S1.idSuc, nombreSuc, PC.idCte, nombreCte, direccionCte

FROM Sucursal **as** S1, Cuenta **as** C, PropietarioCuenta **as** PC, Cliente **as** CL1

WHERE S1.idSuc = C.idSuc

AND C.nroCuenta = PC.nroCuenta

AND PC.idCte = CL1.idCte

AND C.saldo > 0)

)

→ Luego: **SELECT** nombreCte, direccionCte

FROM clientesActivos

WHERE nombreSuc = 'CABA1'

Ej 32)

En forma completa:

Insert into Cuenta

Values (52, 3453000289, 100)

En forma incompleta. Se debe nombrar para que campos se esta dando valores. En este ej, **saldo** quedará con valor nulo si lo permite o con un valor default si está seteado. En otro caso, dará error.

Insert Into Cuenta (idSuc, nroCuenta)

Values (52, 3453000289)

Ej 33)

DELETE FROM cuenta

Where idSuc = 43

and saldo **between** 1000 **and** 5000

Ej 34)

Update cliente

Set direccionCte = 'nuevaDir', ciudadCte = 'nuevaCiudad'

Where idCte = 734

Ej 35)

Update cuenta

Set saldo = saldo * 1.05

Ej 36)

SELECT *

FROM prestamo as P **inner join** propietarioprestamo as PP **on** P.nroPrestamo = PP.nroPrestamo

Se realiza la búsqueda de los número de prestamos de la tabla prestamo, si se encuentra en la otra tabla aparece en el resultado, primero todos los atributos de prestamo y luego todos los de propietario prestamo.

Ej 37)

SELECT *

FROM prestamo as P **left join** propietarioprestamo as PP **on** P.nroPrestamo = PP.nroPrestamo

Ej 38)

SELECT *

FROM prestamo as P **right join** propietarioprestamo as PP **on** P.nroPrestamo = PP.nroPrestamo

Ej 39)

SELECT *

FROM prestamo as P **full join** propietarioprestamo as PP **on** P.nroPrestamo = PP.nroPrestamo

Ej 40)

SELECT *

FROM prestamo as P **natural join** propietarioprestamo as PP **on** P.nroPrestamo = PP.nroPrestamo

INNER JOIN

Prestamo			PropietarioPrestamo	
IdSuc	nroPrestamo	importe	idCte	nroPrestamo
1	123	100.000	18	123
3	321	50.000	28	321
17	456	250.000	38	456
24	654	35.000	48	654

LEFT JOIN

Prestamo			PropietarioPrestamo	
IdSuc	nroPrestamo	importe	idCte	nroPrestamo
1	123	100.000	null	null
3	321	50.000	28	321
17	456	250.000	null	null
24	654	35.000	48	654

RIGHT JOIN

Prestamo			PropietarioPrestamo	
IdSuc	nroPrestamo	importe	idCte	nroPrestamo
null	null	null	18	123
3	321	50.000	28	321
17	456	250.000	38	456
null	null	null	48	654

FULL JOIN

Prestamo			PropietarioPrestamo	
IdSuc	nroPrestamo	importe	idCte	nroPrestamo
null	null	null	18	123
3	321	50.000	28	321
17	456	250.000	null	null
24	654	35.000	48	654

NATURAL JOIN

Prestamo			PropietarioPrestamo
IdSuc	nroPrestamo	importe	idCte
1	123	100.000	18
3	321	50.000	28
17	456	250.000	38
24	654	35.000	48