**Enunciado**

**Ejercicio 1**

1- Crear una BD con 3 tablas (socio, Pais, Provincia)

2- La tabla socio posee los campos (Id\_Socio, Nombre, Apellido, email, Id\_Pais, Id\_provincia)

3- La tabla Pais posee los campos (Id\_pais, Nombre)

4- La tabla Provincia posee los campos (Id\_Provincia, Nombre)

5- Una vez creada la BD mostrar el grafico y las relaciones de las mismas

create database tres\_tablas;

use tres\_tablas;

--Creación de tablas

create table pais(

id\_pais int not null UNIQUE,

nombre varchar(80) not null,

primary key(id\_pais)

);

create table provincia(

id\_provincia int not null,

nombre varchar(80) not null,

primary key(id\_provincia),

);

create table socio(

id\_socio int not null,

nombre varchar(30) not null,

apellido varchar(50) not null,

id\_pais int,

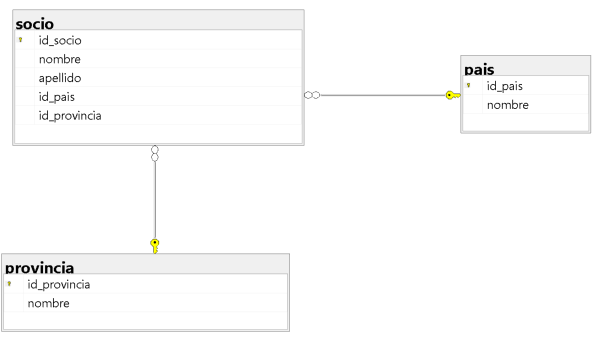
id\_provincia int,

primary key(id\_socio),

foreign key(id\_pais) references pais(id\_pais),

foreign key(id\_provincia) references provincia(id\_provincia)

);



**Ejercicio 2**

1\_ Realizar la consulta de traer todos los datos de la tabla socios, otra de la tabla país y provincia

2\_Realizar una consulta que traiga solos los registros

Id\_socio, nombre, email de los socios de la provincia de Bs AS

3\_ Realizar una consulta que traiga solos los registros

Id\_socio, nombre, email de los socios llamados Pablo

4\_Sacar la cuenta del total de socios, otra para provincias.

5\_ Borrar los registros de Id\_Socio=2

6\_ Cambiar el email del socio llamado X

7\_ Inserte un nuevo país

8\_ inserte un nuevo socio

9\_ Modifique el nombre de un Pais

10\_ Modifique el país de los socios de Nigeria

11\_ saque el promedio de los socios de Argentina

12\_ Muestre un listado de socios agrupados por país en forma ascendente.

--INSERTAR DATOS

--PAIS

insert into pais([id\_pais], [nombre]) values (1, 'Argentina');

insert into pais([id\_pais], [nombre]) values (2, 'Alemania');

insert into pais([id\_pais], [nombre]) values (3, 'Brasil');

insert into pais([id\_pais], [nombre]) values (4, 'República Popular China');

--PROVINCIA

insert into provincia([id\_provincia], [nombre]) values (1, 'Buenos Aires');

insert into provincia([id\_provincia], [nombre]) values (2, 'Rosario');

insert into provincia([id\_provincia], [nombre]) values (3, 'Fujian');

insert into provincia([id\_provincia], [nombre]) values (4, 'Hubei');

insert into provincia([id\_provincia], [nombre]) values (5, 'Berlín');

insert into provincia([id\_provincia], [nombre]) values (6, 'Hamburgo');

insert into provincia([id\_provincia], [nombre]) values (7, 'Bahía');

--SOCIO

insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (1, 'Zoe', 'Perez', 'zoeperez@gmail.com', 1, 1);

insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (2, 'Armin', 'Ajlsld', 'arminvanbuuren@gmail.com', 2, 5);

insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (3, 'Chen', 'Sorktu', 'chenchin@gmail.com', 4, 4);

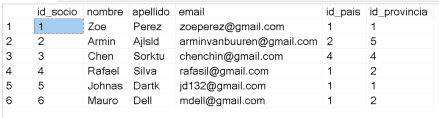
insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (4, 'Rafael', 'Silva', 'rafasil@gmail.com', 1, 2);

insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (5, 'Johnas', 'Dartk', 'jd132@gmail.com', 1, 1);

insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (6, 'Mauro', 'Dell', 'mdell@gmail.com', 1, 2);

1)

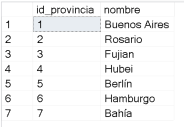
SELECT \* FROM socio;



SELECT \* FROM pais;



SELECT \* FROM provincia;



2)

SELECT s.id\_socio, s.nombre, s.email

FROM socio s

WHERE s.id\_provincia LIKE 1;

3)

SELECT s.id\_socio, s.nombre, s.email

FROM socio s

WHERE s.nombre LIKE 'pablo';

4)

SELECT pr.nombre AS nombre\_socio, COUNT(s.id\_provincia) AS cantidad

FROM socio s

inner join provincia pr on pr.id\_provincia = s.id\_provincia

group by pr.nombre;

5)

DELETE FROM socio

WHERE id\_socio = 1;

6)

UPDATE socio

SET email = 'aajlsd@yahoo.com'

WHERE [nombre] = 'Armin'

8)

insert into socio([id\_socio], [nombre], [apellido], [email], [id\_pais], [id\_provincia]) values (8, 'Claudia', 'Njkrs', 'clud007@gmail.com', 3, 7);

9)

UPDATE socio

SET email = 'aajlsd@yahoo.com'

WHERE [nombre] = 'Armin'

10)

UPDATE p

SET nombre = 'Brasil'

FROM pais p

INNER JOIN socio s

ON p.id\_pais = s.id\_pais

WHERE p.nombre = 'Nigeria'

11)

SELECT (COUNT(s.id\_pais)/8)\*100 AS PROMEDIO

FROM socio s

INNER JOIN pais p

ON p.id\_pais = s.id\_pais

WHERE p.nombre = 'ARGENTINA';

12)

SELECT \*

FROM socio s

ORDER BY s.id\_pais ASC

**Ejercicio 3**

1- Crear una BD con 4 tablas (Juego, Jugadores, Bando, Juego\_Jugadores)

2- La tabla Juego posee los campos (Cod\_Juego, Nombre, Descripción, Cod\_Jugador)

3- La tabla Jugador posee los campos (Cod\_Jugador, Nombre, Apellido, Cod\_Bando, )

4- La tabla Bando posee los campos (Cod\_Bando, Nombre)

5- La tabla intermedia Juego\_jugadores, posee los cod\_juego y los Cod\_Jugadores.

6- Una vez creada la BD mostrar el grafico y las relaciones de las mismas.

create database cuatro\_tablas;

use cuatro\_tablas;

--Creación de tablas

create table bando(

cod\_bando int not null UNIQUE,

nombre varchar(80) not null,

primary key(cod\_bando)

);

create table jugador(

cod\_jugador int not null UNIQUE,

nombre varchar(80) not null,

apellido varchar(80) not null,

cod\_bando int,

primary key(cod\_jugador),

foreign key(cod\_bando) references bando(cod\_bando)

);

create table juego(

cod\_juego int not null UNIQUE,

nombre varchar(80) not null,

descripción varchar(80) not null,

cod\_jugador int,

primary key(cod\_juego),

foreign key(cod\_jugador) references jugador(cod\_jugador)

);

create table intermedia(

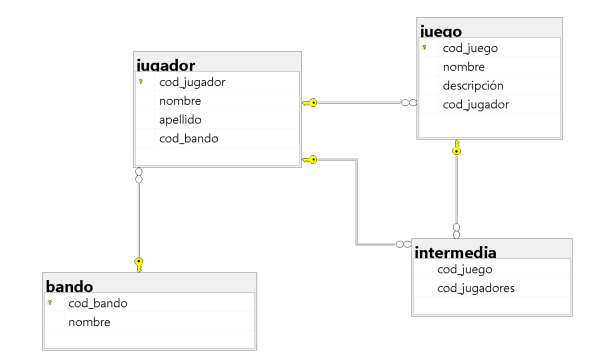
cod\_juego int,

cod\_jugadores int,

foreign key(cod\_juego) references juego(cod\_juego),

foreign key(cod\_jugadores) references jugador(cod\_jugador)

);



**Ejercicio 4**

1\_ Realizar la consulta de traer todos los datos de la tabla Juego, otra de la tabla Jugadores y Bando.

SELECT \* FROM juego;

SELECT \* FROM jugador;

SELECT \* FROM bando;

2\_Realizar una consulta que traiga solos los jugadores de cada juego

SELECT j.\*

FROM jugador j

inner join juego ju

ON ju.cod\_jugador = j.cod\_jugador

ORDER BY ju.cod\_juego asc

3\_ Realizar una consulta que traiga solos los registros de un determinador juego y bando.

SELECT ju.nombre AS juego, ba.nombre AS bando

FROM juego ju

INNER JOIN jugadores j

ON ju.cod\_jugador = j.cod\_jugador

INNER JOIN bando ba

ON j.cod\_bando = ba.cod\_bando

WHERE ju.cod\_juego = 1;

4\_Sacar la cuenta del total de jugadores, y bando de un determinado juego.

SELECT COUNT(ju.cod\_jugador) AS JUGADORES, COUNT(ba.cod\_bando) AS Bando

FROM jugador ju

inner join bando ba

on ba.cod\_bando = ju.cod\_bando

inner join juego j

on j.cod\_juego = ju.cod\_bando

WHERE ju.cod\_jugador = j.cod\_jugador

5\_ Borrar los registros de Id\_juego=2

DELETE FROM juego

WHERE id\_juego = 2;

6\_ Cambiar el nombre del bando código 3.

UPDATE bando

SET nombre = 'Dark HOurs'

WHERE [cod\_bando] = 3

7\_ Inserte un nuevo jugador

insert into jugador([cod\_jugador], [nombre], [apellido], [cod\_bando]) values (8, 'Zoe', 'Perez', 2);

8\_ inserte un nuevo Juego

insert into juego([cod\_juego], [nombre], [descripcion], [cod\_jugador]) values (5, 'Burako', 'El objetivo es ser el primero en llegar a los 3 mil puntos. Se logran formando…', 2);

9\_ Modifique el nombre de un Juego

UPDATE juego

SET nombre = 'Counter Strike'

WHERE [nombre] = 'Counter'

10\_ Modifique el Nombre de uno de los bandos

UPDATE bando

SET nombre = 'Dark HOurs'

WHERE [nombre] = 'Dark #256'

**Ejercicio 5**

Realizar una backup de la BD y guardar las consultas realizadas

**Ejercicio 6**

Agregar más información a una de las tablas y realizar un restore con el backup realizado en el Ejercicio 3.

¿Qué sucedió con la nueva información ingresada?