

Create database IF5100_2023_Proyecto_RaSe

Drop database IF5100_2023_Proyecto_RaSe

use master

USE IF5100_2023_Proyecto_RaSe

--CREACION DE TABLAS

--EVENTOS

Create table RyS_Club(

id_club int identity(1,1) Primary key,

pais varchar(30) not null,

recursos int not null,

ubicacion varchar(100) not null,

id_gerente int --FK

)

Create table RyS_Contabilidad(

id_pago int identity(1,1) PRIMARY KEY,

monto int not null,

fecha_transaccion date not null,

id_club int, --FK

tipo_pago varchar(10), --evento, promocion, plan, otro

descrip_pago varchar (50))

Create table RyS_ContabilidadClub(

id_pago int primary key references RyS_Contabilidad(id_pago),

id_socio int references RyS_SocioDatosI(id_socio),

estado varchar(15) --Cancelado, Pendiente

)

```
Create table RyS_ContabilidadEventoPublico(  
id_pago int primary key references RyS_Contabilidad(id_pago),  
cod_entrada int  
)
```

```
Create table RyS_Contrato(  
nombre_empresa varchar(50) PRIMARY KEY,  
tipo_contrato varchar(30) not null,  
servicio varchar(100) not null,  
inicio_contrato date,  
fin_contrato date,  
id_club int, --FK  
Foreign key(id_club) references RyS_Club(id_club))
```

```
Create table RyS_Empleado(  
id_empleado int identity(1,1) PRIMARY KEY,  
nombre varchar(30) not null,  
apellido varchar(50) not null,  
horario varchar(50) not null,  
pago int not null,  
pago_horas_extra int not null,  
puesto_trabajo varchar(50)not null,  
estado varchar(20) not null,  
id_recurso int, --FK  
Foreign key(id_recurso) references RyS_RecursosClub(id_recurso),  
)
```

```
Create table RyS_Evento_Privado(  
id_socio int not null,
```

```
id_evento int not null,  
asistentes int not null  
PRIMARY KEY(id_socio,id_evento))
```

--Foreign keys

```
Create table RyS_Evento_Publico(  
cod_entrada int identity(1,1) not null,  
id_evento int not null,  
lugar varchar(50) not null,  
precio int not null,  
PRIMARY KEY(cod_entrada,id_evento))
```

```
Create table RyS_Eventos(  
id_evento int identity(1,1) PRIMARY KEY not null,  
nombre varchar(30) not null,  
--Se cambio por int  
id_recurso int not null,  
--Se subió a 3000  
evento varchar(3000) not null,  
tipo varchar(20) not null,  
fecha date not null,  
cant_personas int not null)
```

--Desnormalización 3

--Se crea tabla RyS_EventosRecurso

--Es más funcional para los procedimientos y joins

```
Create table RyS_EventosRecurso(  
id_evento int foreign key references RyS_Eventos(id_evento),
```

id_recurso int foreign key references RyS_RecursosClub(id_recurso))

Create table RyS_Invitado(

id_invitado int identity(1,1) PRIMARY KEY,

nombre varchar(30) not null,

apellido varchar(50) not null,

cedula int not null,

id_socio int, --PK

Foreign key(id_socio) references RyS_SocioDatosI(id_socio))

Create table RyS_Mantenimiento(

id_recurso int,

id_empleado int,

dia date,

area varchar(50),

hora time

--,Primary Key(id_recurso,id_empleado)

)

Create table RyS_PagoHorasExtra(

id_empleado int primary key references RyS_Empleado(id_empleado),

cant_horas int,

dia date,

pago_total int)

Create table RyS_Personal_Evento(

id_personal int,

id_evento int,

actividad varchar(20)

Primary Key(id_personal,id_evento))

```
Create table RyS_Promociones(  
id_promo int identity(1,1) PRIMARY KEY,  
descripcion varchar(100) not null,  
inicio_promo date,  
fin_promo date,  
tipo varchar(20) not null,  
id_club int, --FK  
Foreign key(id_club) references RyS_Club(id_club))
```

```
create table RyS_RecursosClub(  
id_recurso int identity(1,1) primary key,  
sitio varchar(50) not null,  
id_club int, --FK  
ubicacion_club varchar(100) not null,  
categorizacion varchar(30) not null,  
Foreign key(id_club) references RyS_Club(id_club))
```


--Desnormalización 2

--Se agrega la tabla que tiene solo los datos de id_club y el recurso

--Esto es mejor para las consultas para que no se traigan todos los datos

--que no son necesarios

```
Create table RyS_RecursoYClub(  
id_recurso int foreign key references RyS_RecursosClub(id_recurso),  
id_club int foreign key references RyS_Club(id_club)  
)
```

```
create table RyS_ReservaHora(  
id_recurso int foreign key references RyS_RecursosClub(id_recurso),  
hora time,  
dia date,  
numero_reservas int)
```

```
create table RyS_Reserva_Espacio(  
cod_reserva int identity(1,1) Primary Key,  
descrip varchar(200) not null,  
id_recurso int not null, --FK  
id_socio int not null, --FK  
dia date not null,  
hora_i datetime not null,  
hora_f datetime,  
Foreign key(id_recurso) references RyS_RecursosClub(id_recurso),  
Foreign key(id_socio) references RyS_SocioDatosI(id_socio),  
)
```

```
create table RyS_ServiciosAprovechados(  
id_recurso int,  
id_socio int, --FK  
id_empleado int, --FK  
fechas date,  
horas time  
Constraint pk_Servicios Primary key (id_recurso,id_socio),  
Foreign key(id_recurso) references RyS_RecursosClub(id_recurso),  
Foreign key(id_socio) references RyS_SocioDatosI(id_socio),  
Foreign key(id_empleado) references RyS_Empleado(id_empleado)
```

)

```
Create table RyS_Socio(  
id_socio int identity(1,1) PRIMARY KEY,  
nombre varchar(50) not null,  
apellido varchar(50) not null,  
cedula int not null,  
correo varchar(100) not null,  
plan_ofrece varchar(50) not null,  
fecha_afiliacion date)
```

```
Drop table RyS_Socio
```

--Desnormalización 1

--Esto es más práctico porque a la hora de manejar los datos

--Ya que los datos de la tabla RyS_SocioDatosI son los que

--se manejan más frecuentemente

```
Create table RyS_SocioDatosI(  
id_socio int identity(1,1) PRIMARY KEY,  
nombre varchar(50) not null,  
apellido varchar(50) not null,  
cedula int not null,  
plan_ofrece varchar(50) not null)
```

```
Create table RyS_SocioDatosExtra(  
id_socio int primary key references RyS_SocioDatosI(id_socio),  
correo varchar(100) not null,
```

fecha_afiliacion date)

create table RyS_Socio_Club(

id_club int, --FK,

id_socio int, --FK

Foreign key(id_club) references RyS_Club(id_club),

Foreign key(id_socio) references RyS_SocioDatosI(id_socio))

create table RyS_SocioProm(

id_promo int, --FK

id_socio int, --FK

Foreign key(id_socio) references RyS_SocioDatosI(id_socio),

Foreign key(id_promo) references RyS_Promociones(id_promo))

Create table RyS_VacacionesEmpl(

cod_vacaciones int identity(1,1) PRIMARY KEY,

inicio_vac date not null,

fin_vac date not null,

id_empleado int, --FK

Foreign key(id_empleado) references RyS_Empleado(id_empleado))

Insert into RyS_SocioProm values (1,2)

Insert into RyS_Promociones values ('50% de descuento Mes de Julio en el Gimnasio',
'07/01/2023', '07/31/2023', 'Descuento',2)

GO

-- /////Procedimientos Almacenados Basicos///// --

--1 SOCIO

--INSERTAR SOCIO

Create Procedure sp_insert_socio

@nombre varchar(50) ,

@apellido varchar(50) ,

@cedula int ,

@correo varchar(100) ,

@plan_ofrece varchar(50) ,

@id_club int --FK

AS

Begin

Declare @id_socio int

INSERT INTO RyS_SocioDatosI(nombre,apellido,cedula,plan_ofrece)

Values(@nombre,@apellido,@cedula,@plan_ofrece)

Select @id_socio = id_socio from RyS_SocioDatosI where nombre = @nombre

Insert into RyS_SocioDatosExtra(id_socio,correo,fecha_afiliacion) values (@id_socio,
@correo,convert(date, getdate()))

End

```
Exec sp_insert_socio 'Raúl', 'Miranda', 305460445, 'raul.miranda@ucr.ac.cr', 'Básico', 1
```

```
Exec sp_insert_socio 'Fernando', 'Fernandez', 609890986, 'fernando.fernandez@ucr.ac.cr',  
'Completo', 1
```

```
GO
```

```
--SELECCIONAR SOCIO
```

```
Create Procedure sp_select_socio
```

```
@id_socio int
```

```
AS
```

```
Select RyS_SocioDatosI.id_socio,nombre, apellido, cedula, correo, fecha_afiliacion, plan_ofrece
```

```
from RyS_SocioDatosI, RyS_SocioDatosExtra
```

```
where RyS_SocioDatosI.id_socio = @id_socio and RyS_SocioDatosExtra.id_socio = @id_socio
```

```
GO
```

```
exec sp_select_socio 1
```

```
--ACTUALIZAR SOCIO
```

```
Create Procedure sp_update_socio(
```

```
@id_socio int,
```

```
@nombre varchar(50),
```

```
@apellido varchar(50),
```

```
@cedula int ,
```

```
@correo varchar(100),
```

```
@plan_ofrece varchar(50),
```

```
@fecha_afiliacion date)
```

AS

Begin

UPDATE RyS_SocioDatosI SET

nombre=@nombre,

apellido=@apellido,

cedula =@cedula,

plan_ofrece = @plan_ofrece

where id_socio=@id_socio

Update RyS_SocioDatosExtra set

correo=@correo,

fecha_afiliacion=@fecha_afiliacion

where id_socio=@id_socio

End

Exec sp_update_socio 2,'Ferndando', 'Fernandez', 609890986, 'fernando.fernandez@ucr.ac.cr',
'Completo', '06/21/2023'

--ELIMINAR SOCIO

Create Procedure sp_delete_socio

@id_socio int

AS

Delete from RyS_Socio where id_socio = @id_socio

GO

--2 EVENTO

--INSERTAR EVENTO

Create Procedure sp_insert_eventos

```
@nombre varchar(30) ,
@id_recurso varchar(100) ,
@evento varchar(300) ,
@tipo varchar(20),
@fecha date,
@cant_personas int
AS
INSERT INTO RyS_Eventos(nombre,id_recurso,evento,tipo,fecha,cant_personas)
Values(@nombre,@id_recurso,@evento,@tipo,@fecha,@cant_personas)
GO
```

```
exec sp_insert_eventos 'Fiesta en la piscina', 1 , 'Se realizara una fiesta en la ',
'Recreativo','06-30-2023' , 100
```

```
--SELECCIONAR EVENTO
Create Procedure sp_select_evento
@id_evento int
AS
Select * from RyS_Eventos
where id_evento = @id_evento
GO
```

```
sp_select_evento 2
```

```
--UPDATE EVENTO
Create Procedure sp_update_eventos
@id_evento int,
@nombre varchar(30),
@id_recurso varchar(100),
```

```

@evento varchar(3000),
@tipo varchar(20),
@fecha date ,
@cant_personas int
AS
UPDATE RyS_Eventos SET
nombre=@nombre,
id_recurso=@id_recurso,
evento=@evento,
tipo=@tipo,
fecha=@fecha,
cant_personas=@cant_personas
where id_evento=@id_evento
GO

```

```

exec sp_update_eventos 2, 'Fiesta en la piscina', 2 , 'No se realizara una fiesta ',
'Recreativo','12-01-1980' , 100

```

--ELIMINAR EVENTO

```

Create Procedure sp_delete_evento

```

```

@id_evento int

```

```

AS

```

```

Delete from RyS_Eventos where @id_evento = @id_evento

```

```

GO

```

--3 CLUB

-----Funciona-----

--INSERTAR CLUB

Create procedure sp_insert_club

@pais varchar(30),

@recursos int,

@ubicacion varchar(100),

@id_gerente int --FK

As

INSERT INTO RyS_Club(pais,recursos,ubicacion,id_gerente)

VALUES(@pais,@recursos,@ubicacion,@id_gerente)

GO

Drop procedure sp_insert_club

exec sp_insert_club 'Costa Rica', 0, 'Dulce Nombre, Cartago', 1

-----Funciona-----

--SELECCIONAR CLUB

Create Procedure sp_select_club

@id_club int

AS

Select * from RyS_Club

where id_club = @id_club

GO

exec sp_select_club 2

-----Funciona-----

--UPDATE CLUB

Create procedure sp_update_club

@id_club int,

@pais varchar(30),

@recursos int,

@ubicacion varchar(100),

@id_gerente int

AS

UPDATE RyS_Club SET

pais=@pais,

recursos=@recursos,

ubicacion=@ubicacion,

id_gerente=@id_gerente

where id_club=@id_club

GO

exec sp_update_club 1, 'Costa Rico', 2, 'Paraíso, Cartago', 2

Drop procedure sp_update_club

-----Funciona-----

--ELIMINAR CLUB

Create Procedure sp_delete_club

@id_club int

AS

Delete from RyS_Club where @id_club = @id_club

GO

exec sp_delete_club 1

--4 EMPLEADO

--INSERTAR EMPLEADO

Create Procedure sp_insert_empleado(

@nombre varchar(30),

@apellido varchar(50),

@horario varchar(20),

@pago int ,

@pago_horas_extra int,

@puesto_trabajo varchar(50),

@estado varchar(20),

@id_recurso int)

AS

Begin

INSERT INTO

RyS_Empleado(nombre,apellido,horario,pago,pago_horas_extra,puesto_trabajo,estado,id_recurso
)

VALUES(@nombre,@apellido,@horario,@pago,@pago_horas_extra,@puesto_trabajo,@estado,@i
d_recurso)

End

GO


```
exec sp_insert_empleado 'Mauricio', 'Alvarado', 'L-V: 9:00am-5:00pm S-D:7:00am-12:00pm', 1000,  
1500, 'Limpieza', 'Activo', 1
```

```
--SELECCIONAR EMPLEADO
```

```
Create Procedure sp_select_empleado
```

```
@id_empleado int
```

```
AS
```

```
Select * from RyS_Empleado
```

```
where id_empleado = @id_empleado
```

```
GO
```

```
--ACTUALIZAR EMPLEADO
```

```
Create Procedure sp_update_empleado(
```

```
@id_empleado int,
```

```
@nombre varchar(30),
```

```
@apellido varchar(50) ,
```

```
@horario varchar(20) ,
```

```
@pago int,
```

```
@pago_horas_extra int ,
```

```
@puesto_trabajo varchar(50),
```

```
@estado varchar(20),
```

```
@id_recurso int )
```

```
AS
```

```
Begin
```

```
UPDATE RyS_Empleado SET
```

```
nombre=@nombre,
```

```
apellido=@apellido,
```

```
horario=@horario,  
pago=@pago,  
pago_horas_extra=@pago_horas_extra,  
puesto_trabajo=@puesto_trabajo,  
estado=@estado,  
id_recurso=@id_recurso  
where id_empleado= @id_empleado  
End  
GO  
  
exec sp_update_empleado 3,'Sharon', 'Hernandez', 'L-J: 10:00pm-6:00am', 1500, 2250,  
'Seguridad', 'Activo',2
```

--ELIMINAR CLUB

Create Procedure sp_delete_empleado

@id_empleado int

AS

Delete from RyS_Empleado where @id_empleado = @id_empleado

GO

--5 RECURSOS_CLUB

--INSERTAR RECURSOS_CLUB

create procedure sp_insertar_recursos_club(

@sitio varchar(50),

@id_club int, --FK

@ubicacion_club varchar(100),

@categorizacion varchar(30))

AS

Begin

INSERT INTO RyS_RecursosClub(sitio,id_club,ubicacion_club,categorizacion)

VALUES(@sitio,@id_club,@ubicacion_club,@categorizacion)

End

Go

exec sp_insertar_recursos_club 'Piscina',2,'Área Recreativa', 'Recreación'

exec sp_insertar_recursos_club 'Salon',2,'Área para actividades', 'Actividades'

--SELECCIONAR RECURSOS_CLUB

Create Procedure sp_select_recursos_club(

@id_recurso int

)

AS

Select * from RyS_RecursosClub

where id_recurso = @id_recurso

GO

Drop procedure sp_select_recursos_club

exec sp_select_recursos_club 1

--UPDATE RECURSOS_CLUB

create procedure sp_update_recursos_club(

@id_recurso int,

@sitio varchar(50),

@id_club int, --FK

@ubicacion_club varchar(100),

```
@categorizacion varchar(30))
```

```
AS
```

```
Begin
```

```
UPDATE RyS_RecursosClub SET
```

```
sitio=@sitio,
```

```
id_club=@id_club,
```

```
ubicacion_club=@ubicacion_club,
```

```
categorizacion=@categorizacion
```

```
where id_recurso=@id_recurso
```

```
End
```

```
GO
```

```
exec sp_update_recursos_club 2,'Piscina',1,'Área Recreativa', 'Diversión'
```

```
--ELIMINAR RECURSOS_CLUB
```

```
Create Procedure sp_delete_recursos_club(
```

```
@id_recurso int )
```

```
AS
```

```
Begin
```

```
Delete from RyS_RecursosClub where @id_recurso = id_recurso
```

```
End
```

```
GO
```

```
--6 EMPLEADO
```

```
--INSERTAR INVITADO
```

```
Create procedure sp_insert_invitado(
```

```
@nombre varchar(30),
@apellido varchar(50),
@cedula int,
@id_socio int --PK
)
AS
Begin
INSERT INTO RyS_Invitado(nombre,apellido,cedula,id_socio)
VALUES(@nombre,@apellido,@cedula,@id_socio)
End
```

```
exec sp_insert_invitado 'Sara', 'Camacho', 304560345,1
```

```
--SELECCIONAR Invitado
Create Procedure sp_select_invitado
@id_invitado int
AS
Select * from RyS_Invitado
where id_invitado = @id_invitado
GO
```

```
exec sp_select_invitado 1
```

```
--UPDATE Invitado
Create procedure sp_update_invitado(
@id_invitado int,
@nombre varchar(30),
@apellido varchar(50) ,
@cedula int,
```

```
@id_socio int --PK
```

```
)
```

```
AS
```

```
Begin
```

```
UPDATE RyS_Invitado SET
```

```
nombre=@nombre,
```

```
apellido=@apellido,
```

```
cedula=@cedula,
```

```
id_socio=@id_socio
```

```
where id_invitado=@id_invitado
```

```
End
```

```
GO
```

```
--ELIMINAR EMPLEADO
```

```
Create Procedure sp_delete_invitado
```

```
@id_invitado int
```

```
AS
```

```
Delete from RyS_Invitado where id_invitado = @id_invitado
```

```
GO
```

```
--Procedimientos complejos
```

```
drop procedure ProcedimientoCursor
```

```
exec ProcedimientoCursor
```

```
--Ciclo con cursor
```

```
Create Procedure ProcedimientoCursor
```

```
As
```

```

Begin
Declare @id_promo int
Begin transaction
Begin try
    Declare finPromo cursor for
    Select id_promo from RyS_Promociones
    where DateDIFF(month, getdate(),fin_promo) <= 1;--Donde falte 1 mes para que se venza

Open finPromo
Fetch Next from finPromo into @id_promo

While @@FETCH_STATUS = 0
Begin
Select id_socio from RyS_SocioProm where id_promo =@id_promo
Fetch Next From finPromo into @id_promo

End
Close finPromo
Deallocate finPromo
End try
Begin Catch
    SELECT      --- entró por un error que activó el CATCH
    ERROR_NUMBER() AS ErrorNumber
    ,ERROR_SEVERITY() AS ErrorSeverity
    ,ERROR_STATE() AS ErrorState
    ,ERROR_PROCEDURE() AS ErrorProcedure
    ,ERROR_LINE() AS ErrorLine
    ,ERROR_MESSAGE() AS ErrorMessage;

```

```
IF @@TRANCOUNT > 0 -- si la transacción quedó abierta por lo que el TRANCOUNT es mayor a
cero
```

```
    ROLLBACK TRANSACTION;
```

```
END CATCH;
```

```
IF @@TRANCOUNT > 0 -- no hubo error por lo tanto debe concluir la transacción con un commit.
```

```
    COMMIT TRANSACTION;
```

```
End
```

```
Go
```

```
Select * from RyS_Eventos
```

```
Select * from RyS_SocioDatosI
```

```
Select * from RyS_Evento_Privado
```

```
Insert into RyS_Evento_Privado values(1,1,5)
```

```
--Ciclo con tabla temporal
```

```
--Este obtiene el idSocio y los asistentes cuando hay un evento privado
```

```
Drop Procedure ProdTablaTemporal
```

```
Create Procedure ProdTablaTemporal
```

```
@id_evento int
```

```
As
```


Begin

Begin transaction

Begin try

DECLARE @AsistentesEvento TABLE (id_socio int, asistentes int)

Declare @idSocio int,@cedula int,@asistentesSocio int, @nombre varchar(50)

INSERT INTO @AsistentesEvento SELECT id_socio, asistentes

FROM RyS_Evento_Privado where id_evento = @id_evento

While exists (Select * from @AsistentesEvento)

Begin

set rowcount 1

Select @idSocio = id_socio,@asistentesSocio = asistentes from @AsistentesEvento

Select @cedula = cedula, @nombre = nombre from RyS_SocioNuevo where id_socio = @idSocio

Delete @AsistentesEvento where id_Socio = @idSocio

Print Concat(@cedula,'|',@nombre, '|' , @asistentesSocio)

set rowcount 0

End

End try

Begin Catch

SELECT --- entró por un error que activó el CATCH

ERROR_NUMBER() AS ErrorNumber

,ERROR_SEVERITY() AS ErrorSeverity

,ERROR_STATE() AS ErrorState

,ERROR_PROCEDURE() AS ErrorProcedure

,ERROR_LINE() AS ErrorLine

```
,ERROR_MESSAGE() AS ErrorMessage;
```

```
IF @@TRANCOUNT > 0 -- si la transacción quedó abierta por lo que el TRANCOUNT es mayor a  
cero
```

```
ROLLBACK TRANSACTION;
```

```
END CATCH;
```

```
IF @@TRANCOUNT > 0 -- no hubo error por lo tanto debe concluir la transacción con un commit.
```

```
COMMIT TRANSACTION;
```

```
End
```

```
Go
```

```
exec ProdTablaTemporal 2
```

```
--Ciclo con variable tipo tabla
```

```
--Las reservas de las áreas de la semana que viene
```

```
Select * from RyS_Reserva_Espacio
```

```
Select * from RyS_Mantenimiento
```

```
Select * from RyS_SocioDatosI
```

```
Select * from RyS_RecursosClub
```

```
Select * from RyS_Empleado
```

```
Select * from RyS_RecursoYClub
```

```
Select * from RyS_Club
```

```
Insert into RyS_Reserva_Espacio values('Uso de la piscina', 1,1,'06/26/2023','8:00','10:00' )
```

```
Insert into RyS_Reserva_Espacio values('Uso de la piscina', 1,2,'06/25/2023','9:00','10:00' )
```

Insert into RyS_Mantenimiento values(1,2,'06/25/2023',1,'5:00')

Insert into RyS_RecursoYClub values(1,2)

Drop procedure ProcedimientoVariableTipoTabla

Create Procedure ProcedimientoVariableTipoTabla

@id_club int

As

Begin

Declare @dia date, @idRecurso int, @idAprovecha int, @idRecursosClub int, @hora time

Begin transaction

Begin try

Select @idRecursosClub = id_recurso from RyS_RecursoYClub where id_club = @id_club

SELECT * INTO #Reservas FROM RyS_Reserva_Espacio

where DateDIFF(DAY, getdate(),dia) <= 7 and id_recurso = @idRecursosClub;

SELECT * INTO #Mantenimiento FROM RyS_Mantenimiento

where DateDIFF(DAY, getdate(),dia) <= 7 and id_recurso = @idRecursosClub;

While Exists (Select * from #Reservas)

Begin

set rowcount 1

Select @dia = dia, @idRecurso = id_recurso, @idAprovecha = id_socio, @hora = hora_i from #Reservas

Print concat('El día: ', @dia, ' se va utilizar ', @idRecurso, ' por el socio ', @idAprovecha, ' a la hora ', convert (time,@hora))

Delete #Reservas where id_recurso = @idRecurso

set rowcount 0

End

While Exists (Select * from #Mantenimiento)

Begin

set rowcount 1

Select @dia = dia, @idRecurso = id_recurso, @idAprovecha = id_empleado, @hora = hora from #Mantenimiento

Print concat('El día:', @dia, ' se le va a dar mantenimiento ', @idRecurso, ' por el empleado ', @idAprovecha, ' a la hora ', convert (time,@hora));

Delete #Mantenimiento where id_recurso = @idRecurso

set rowcount 0

End

DROP TABLE #Reservas

DROP TABLE #Mantenimiento

End try

Begin Catch

SELECT --- entró por un error que activó el CATCH

ERROR_NUMBER() AS ErrorNumber

```
,ERROR_SEVERITY() AS ErrorSeverity  
,ERROR_STATE() AS ErrorState  
,ERROR_PROCEDURE() AS ErrorProcedure  
,ERROR_LINE() AS ErrorLine  
,ERROR_MESSAGE() AS ErrorMessage;
```

```
IF @@TRANCOUNT > 0 -- si la transacción quedó abierta por lo que el TRANCOUNT es mayor a  
cero
```

```
ROLLBACK TRANSACTION;
```

```
END CATCH;
```

```
IF @@TRANCOUNT > 0 -- no hubo error por lo tanto debe concluir la transacción con un commit.
```

```
COMMIT TRANSACTION;
```

```
End
```

```
Go
```

```
exec ProcedimientoVariableTipoTabla 2
```

```
--Procedimiento para reporte con tabla
```

```
--Obtener los empleados que van a salir a vacaciones el proximo mes
```

```
Select * from RyS_Empleado
```

```
Select * from RyS_VacacionesEmpl
```

```
Insert into RyS_Empleado values ('Sharon', 'Hernandez', 'L-J: 10:00pm-6:00am', 1500, 2250,  
'Seguridad', 'Activo',2)
```

```
Insert into RyS_VacacionesEmpl values ('06/28/2023', '07/05/2023',2)
```

```
Insert into RyS_VacacionesEmpl values ('07/01/2023', '07/31/2023',3)
```

```
Delete from RyS_VacacionesEmpl where id_empleado = 3
```

--Procedimiento para reporte con tabla

--Obtener los empleados que van a salir a vacaciones el proximo mes

Drop procedure ProdVacacionesEmple

Create Procedure ProdVacacionesEmple

As

Begin

Begin transaction

Begin try

Declare @id_empleado int,@nombEmpl varchar(50), @inicioVaC DATE ,@diasVac int,@mesVac int;

DECLARE @Vacaciones table(id_empleado int, nombre varchar(50), inicioVac date, finVac date);

--Declare @id_recurso int

--Select @id_recurso = id_recurso from RyS_RecursoYClub where id_club = 2

INSERT INTO @Vacaciones SELECT distinct RyS_Empleado.id_empleado, RyS_Empleado.nombre,

RyS_VacacionesEmpl.inicio_vac,RyS_VacacionesEmpl.fin_vac

FROM RyS_VacacionesEmpl,RyS_Empleado, RyS_RecursoYClub where DateDIFF(DAY,
getdate(),inicio_vac) <= 31

and RyS_VacacionesEmpl.id_empleado = RyS_Empleado.id_empleado and

RyS_RecursoYClub.id_recurso = RyS_Empleado.id_recurso

and RyS_RecursoYClub.id_club = 2;

--and DateDIFF(MONTH, getdate(),inicio_vac) <= 1

```

--While exists (Select * from @Vacaciones)

--Begin

    ---set rowcount 1

    Select id_empleado, nombre, inicioVac, DATEDIFF (DAY, inicioVac, finVac) as DiasVacaciones
    from @Vacaciones

    --Print concat(@id_empleado , '|' , @nombEmpl , '|' Se va de vacaciones el:' , @inicioVaC,
    '|' Días: ' , @diasVac)

    --Delete @Vacaciones where id_empleado = @id_empleado

    --set rowcount 0

--End

```

End try

Begin Catch

```

    SELECT      --- entró por un error que activó el CATCH
    ERROR_NUMBER() AS ErrorNumber
    ,ERROR_SEVERITY() AS ErrorSeverity
    ,ERROR_STATE() AS ErrorState
    ,ERROR_PROCEDURE() AS ErrorProcedure
    ,ERROR_LINE() AS ErrorLine
    ,ERROR_MESSAGE() AS ErrorMessage;

    IF @@TRANCOUNT > 0 -- si la transacción quedó abierta por lo que el TRANCOUNT es mayor a
    cero

    ROLLBACK TRANSACTION;

```

END CATCH;

IF @@TRANCOUNT > 0 -- no hubo error por lo tanto debe concluir la transacción con un commit.

COMMIT TRANSACTION;

End

Go

exec ProdVacacionesEmple

--El de reporte sera el de para decir cuantos empleados hay por departamento

Drop procedure EmpleadosCadaArea

Create Procedure EmpleadosCadaArea

@id_Club int

AS

Begin

Begin Transaction

Begin try

Declare @cant_empleados int, @id_recurso_actual int

SELECT distinct id_recurso into #Recursos FROM RyS_RecursosClub where id_club = @id_club


```
Select RyS_Empleado.id_empleado, RyS_Empleado.id_recurso into #Empleados
from RyS_Empleado, RyS_Club, RyS_RecursosClub where RyS_RecursosClub.id_club = @id_club
AND RyS_RecursosClub.id_recurso = RyS_Empleado.id_recurso
And RyS_Empleado.estado = 'Activo'
```

```
While Exists (Select * from #Recursos)
```

```
Begin
```

```
    set rowcount 1
        Select @id_recurso_actual = id_recurso from #Recursos
    set rowcount 0
        Select @cant_empleados = count(id_recurso) from #Empleados where id_recurso
= @id_recurso_actual
        Print concat('El área:', @id_recurso_actual , ' cuenta con ' , @cant_empleados , '
empleados ' )
        Delete from #Empleados where id_recurso = @id_recurso_actual
        Delete from #Recursos where id_recurso = @id_recurso_actual
```

```
End
```

```
DROP TABLE #Empleados
```

```
Drop table #Recursos
```

```
End Try
```

```
Begin Catch
```

```
SELECT      --- entró por un error que activó el CATCH
ERROR_NUMBER() AS ErrorNumber
,ERROR_SEVERITY() AS ErrorSeverity
,ERROR_STATE() AS ErrorState
,ERROR_PROCEDURE() AS ErrorProcedure
,ERROR_LINE() AS ErrorLine
,ERROR_MESSAGE() AS ErrorMessage;
```

```
IF @@TRANCOUNT > 0 -- si la transacción quedó abierta por lo que el TRANCOUNT es mayor a
cero
```

```
ROLLBACK TRANSACTION;
END CATCH;
```

```
IF @@TRANCOUNT > 0 -- no hubo error por lo tanto debe concluir la transacción con un commit.
```

```
COMMIT Transaction
END
```

Exec EmpleadosCadaArea 2

--Procedimiento para conocer cuantas personas estan trabajando en cada actividad
--especifica cuando se realiza un evento

```
Select * from RyS_Empleado
Select * from RyS_Personal_Evento
Select * from RyS_Eventos
Select * from RyS_RecursosClub
```

```
Insert into RyS_Empleado values ('Sebastian', 'Monge', 'L-V: 8:00am-5:00pm', 1500, 2250,
'Bartender', 'Evento',2)
```

Insert into RyS_Empleado values ('Carlos', 'Carvajal', 'L-V: 8:00am-5:00pm', 1500, 2250, 'Animador', 'Evento', 2)

Insert into RyS_Personal_Evento values (4, 1, 'Bebidas')

Insert into RyS_Personal_Evento values (6, 1, 'Animación')

Drop procedure ProdCategoriaPersonalEvento

Create Procedure ProdCategoriaPersonalEvento

@id_evento int

As

Begin

Begin transaction

Declare @actividad varchar(10), @cantidad int, @actividad_actual varchar(10)

SELECT * INTO #PersonalEvento FROM RyS_Personal_Evento where id_evento = @id_evento

Select DISTINCT actividad into #ActividadesEvento from #PersonalEvento

Begin try

Print concat ('Para el evento: ', @id_evento)

While exists (Select * from #ActividadesEvento)

Begin

set rowcount 1

Select @actividad_actual = actividad from #ActividadesEvento

set rowcount 0

set @cantidad = (Select Count(actividad) from #PersonalEvento where actividad = @actividad_actual)

Print concat (@cantidad , ' empleados estan en ', @actividad_actual)

Delete from #ActividadesEvento where actividad = @actividad_actual

Delete from #PersonalEvento where actividad = @actividad_actual

End

Drop table #ActividadesEvento

Drop table #PersonalEvento

End try

Begin Catch

SELECT

ERROR_NUMBER() AS ErrorNumber

,ERROR_SEVERITY() AS ErrorSeverity

,ERROR_STATE() AS ErrorState

,ERROR_PROCEDURE() AS ErrorProcedure

,ERROR_LINE() AS ErrorLine

,ERROR_MESSAGE() AS ErrorMessage;

IF @@TRANCOUNT > 0

ROLLBACK TRANSACTION;

END CATCH;

```
IF @@TRANCOUNT > 0
```

```
    COMMIT TRANSACTION;
```

```
End
```

```
exec ProdCategoriaPersonalEvento 1
```

```
--Triggers
```

```
--Instead of
```

```
--Verifica congruencia cuando se cree el evento
```

```
--Que se ubique en un área que exista y que sea en una fecha valida
```

```
Drop trigger TrigrEvento
```

```
Create trigger TrigrEvento
```

```
on RyS_Eventos
```

```
Instead of insert
```

```
As
```

```
Begin
```

```
Begin try
```

```
Declare @id_evento int, @nombre varchar(30),
```

```
@id_Recurso int,@evento varchar(50),
```

```
@tipo varchar(20), @fecha date,
```

```
@cant_personas int
```

```
Select @id_evento = id_evento, @nombre = nombre, @id_Recurso = id_recurso, @evento = evento, @tipo = tipo
```

```
, @fecha = fecha, @cant_personas = cant_personas from inserted
```

```
--Verifica que si existe
```

```
IF EXISTS(Select * from RyS_RecursosClub WHERE id_recurso = @id_Recurso)
```

```
Begin
```

```
Insert into RyS_Eventos values (@nombre,@id_Recurso, @evento,  
@tipo,@fecha,@cant_personas)
```

```
End
```

```
End Try
```

```
Begin Catch
```

```
PRINT 'hubo error';
```

```
SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;
```

```
RAISERROR ('HUBO ERROR', 16, 1);
```

```
End catch
```

```
End
```

```
--After trigger
```

```
--Cambia el estado de ese empleado cuando se encuentra en un evento
```

```
Create trigger TrigrPersonalEVENTO
```

```
on RyS_Personal_Evento
```

```
After insert
```

```
As
```

```
Begin
```

Begin try

Declare @id_personal int

Select @id_personal = id_personal from inserted

--Verifica que si existe

IF EXISTS(Select * from inserted)

Begin

Update RyS_Empleado set estado = 'Evento' where id_empleado = @id_personal;

End

End Try

Begin Catch

PRINT 'hubo error';

SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;

RAISERROR ('HUBO ERROR', 16, 1);

End catch

End

Go

--2 After trigger con las 3

--AfterTrigger

Create trigger TrigrManejoEventos

on RyS_Eventos

After insert,delete, update

As

Begin

Begin try

Declare @id_Recurso int, @date date, @id_evento int,@id_personal int

IF exists(Select * from inserted) and not exists(Select * from deleted)

Begin

Select @id_Recurso = id_recurso, @date = fecha from Inserted

--Devuelve todos las reservas que se deben cambiar por el evento

Select * from RyS_Reserva_Espacio where id_recurso = @id_Recurso and dia = @date

End

ELSE IF not exists(Select * from inserted) and exists(Select * from deleted)

Begin

Select @id_evento = id_evento from Inserted

Select * into #PersonalE from RyS_Personal_Evento where id_evento = @id_evento

--Elimina los empleados que estaban asociados a ese evento

While exists (Select * from #PersonalE)

Begin

set rowcount 1

Select @id_personal = id_personal from #PersonalE

Update RyS_Empleado set estado = 'Activo' where id_empleado = @id_personal

Delete from RyS_Personal_Evento where id_personal = @id_personal

Delete from #PersonalE where id_personal = @id_personal

set rowcount 0

End

End

End Try

Begin Catch

PRINT 'hubo error';

SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;

RAISERROR ('HUBO ERROR', 16, 1);

End Catch

End

--AfterTrigger2

--Afecta todos los valores a los que se les asocie el valor socio

DROP TRIGGER UserTrigger

Create trigger UserTrigger

on RyS_SocioDatosI

After insert,delete, update

As

Begin

Begin try

Declare @id_socio int,@idPromocio int, @nombre varchar(50),@apellidos varchar(50),

@idRecurso int, @plan varchar(50)

--Verifica que si existe

IF EXISTS(Select * from inserted) and not exists (select * from deleted)

Begin

 Select @nombre = nombre, @apellidos = apellido from inserted

 Print concat('Ya estas dentro del club, Bienvenido! ', @nombre, ' ', @apellidos)

End

else if not EXISTS(Select * from inserted) and exists (select * from deleted)

begin

 Select @id_socio = id_socio from deleted

 Delete from RyS_SocioDatosI where id_socio = @id_socio

 Delete from RyS_Invitado where id_socio = @id_socio

 Delete from RyS_SocioDatosExtra where id_socio = @id_socio

 Delete from RyS_ServiciosAprovechados where id_socio = @id_socio

 Delete from RyS_Reserva_Espacio where id_socio = @id_socio

 Delete from RyS_Evento_Privado where id_socio = @id_socio

end else begin

 Select @id_socio = id_socio, @plan = plan_ofrece from inserted

 if update (plan_ofrece)

 begin

 if (@plan = 'Básico')

 update RyS_ServiciosAprovechados set

 id_empleado = null, fechas = null, horas = null

 where id_socio = @id_socio

 end

end

End Try

Begin Catch

PRINT 'hubo error';

SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;

RAISERROR ('HUBO ERROR', 16, 1);

End catch

End

Drop trigger AfterRecursosClub

Create trigger AfterRecursosClub

on RyS_RecursosClub

After insert,delete

As

Begin

Begin try

Declare @id_club int,@id_recurso int

--Verifica si se inserto

IF EXISTS(Select * from inserted) and not exists (Select * from deleted)

Begin

Select @id_club = id_club,@id_recurso = id_recurso from inserted

Insert into RyS_RecursoYClub values (@id_recurso,@id_club)

Update RyS_Club set recursos = recursos+1 where id_club = @id_club

End

ELSE

Begin

Select @id_club = id_club,@id_recurso = id_recurso from deleted

Delete from RyS_RecursoYClub where id_club = @id_club and id_recurso = @id_recurso

Update RyS_Club set recursos = recursos-1 where id_club = @id_club

END

End Try

Begin Catch

PRINT 'hubo error';

SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;

RAISERROR ('HUBO ERROR', 16, 1);

End catch

End

Go

Drop trigger ReservasDiaHora

Create trigger ReservasDiaHora

on RyS_Reserva_Espacio

instead of insert,delete

As

Begin

Begin try

Declare @descrip varchar(100),@id_recurso int,@idsocio int,@día date, @horai time, @horaF
time, @id_reserva int

--Verifica si se inserto

IF EXISTS(Select * from inserted) and not exists (Select * from deleted)

Begin

Select @descrip = descrip , @día= dia,@id_recurso = id_recurso, @horai = hora_i, @horaF = hora_f, @idsocio = id_socio from inserted

If NOT EXISTS(Select * from RyS_ReservaHora WHERE dia = @día and hora = @horai and id_recurso = @id_recurso)

Begin

insert into RyS_Reserva_Espacio values
(@descrip,@id_recurso,@idsocio,@día,@horai,@horaF)

Insert into RyS_ReservaHora values (@id_recurso,@horai,@día,1)

End

ELSE

Begin

If exists(Select * from RyS_ReservaHora WHERE dia = @día and hora = @horai and numero_reservas < 15 and id_recurso = @id_recurso)

Begin

Update RyS_ReservaHora set numero_reservas = numero_reservas+1

insert into RyS_Reserva_Espacio values
(@descrip,@id_recurso,@idsocio,@día,@horai,@horaF)

END

else

Print concat('Ya no se pueden insertar más reservaciones', ' ')

END

End

else

begin

```
Select @id_reserva = cod_reserva, @id_recurso = id_recurso, @horai = hora_i from  
deleted
```

```
Delete from RyS_Reserva_Espacio where cod_reserva = @id_reserva
```

```
Update RyS_ReservaHora set numero_reservas = numero_reservas-1 where id_recurso =  
@id_recurso and hora = @horai and dia = @día
```

```
end
```

```
End Try
```

```
Begin Catch
```

```
PRINT 'hubo error';
```

```
SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;
```

```
RAISERROR ('HUBO ERROR', 16, 1);
```

```
End catch
```

```
End
```

```
Go
```

```
Drop trigger MantenimientoDiaHora
```

```
Create trigger MantenimientoDiaHora
```

```
on RyS_Mantenimiento
```

```
INSTEAD OF insert,delete
```

```
As
```

```
Begin
```

```
Begin try
```

```
Declare @id_recurso int, @id_empleado int,@area varchar(50),@día date, @hora time
```

--Verifica si se inserto

IF EXISTS(Select * from inserted) and not exists (Select * from deleted)

Begin

Select @id_empleado = id_empleado , @día = dia,@id_recurso = id_recurso, @hora = hora, @area
= area from inserted

If NOT EXISTS(Select * from RyS_ReservaHora WHERE dia = @día and hora = @hora)

Begin

Insert into RyS_Mantenimiento values (@id_recurso,@id_empleado,@día,@area,@hora)

Insert into RyS_ReservaHora values (@id_recurso,@hora,@día,15)

End

ELSE

Begin

Print concat('Se debe hacer mantenimiento en otro momento', ' ')

END

End

else begin

Select @id_empleado = id_empleado , @día = dia,@id_recurso = id_recurso, @hora = hora
from deleted

Delete from RyS_Mantenimiento where id_empleado = @id_empleado and dia = @día and
@id_recurso = id_recurso and hora = @hora

Update RyS_ReservaHora set numero_reservas = 0 where id_recurso = @id_recurso and
hora = @hora and dia = @día

end

End Try

Begin Catch

PRINT 'hubo error';

SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;

```

        RAISERROR ('HUBO ERROR', 16, 1);
    End catch

End

Go

Drop trigger ReservasServiciosAprovechados

Create trigger ReservasServiciosAprovechados
on RyS_ServiciosAprovechados
instead of insert,delete
As
Begin

Begin try

Declare @id_recurso int,@idsocio int,@idempleado int,@dia date, @hora time, @id_reserva int

--Verifica si se inserto
IF EXISTS(Select * from inserted) and not exists (Select * from deleted)
Begin

Select @dia= fechas,@id_recurso = id_recurso, @hora = horas, @idsocio = id_socio, @idempleado
= id_empleado from inserted

If NOT EXISTS(Select * from RyS_ReservaHora WHERE dia = @dia and hora = @hora and id_recurso
= @id_recurso)
Begin

        insert into RyS_ServiciosAprovechados values
(@id_recurso,@idsocio,@idempleado,@dia,@hora)

        Insert into RyS_ReservaHora values (@id_recurso,@hora,@dia,1)

End

ELSE

```


Begin

 If exists(Select * from RyS_ReservaHora WHERE dia = @dia and hora = @hora and
 numero_reservas < 15 and id_recurso = @id_recurso)

 Begin

 Update RyS_ReservaHora set numero_reservas = numero_reservas+1

 insert into RyS_ServiciosAprovechados values
 (@id_recurso,@idsocio,@idempleado,@dia,@hora)

 END

 else

 Print concat('Ya no se pueden insertar más reservaciones', ' ')

END

End

else

begin

 Select @dia= fechas,@id_recurso = id_recurso, @hora = horas, @idsocio = id_socio,
 @idempleado = id_empleado from deleted

 Delete from RyS_ServiciosAprovechados where id_recurso = @id_recurso and id_socio =
 @idsocio and id_empleado = @idempleado

 and fechas = @dia and horas = @hora

 Update RyS_ReservaHora set numero_reservas = numero_reservas-1 where id_recurso =
 @id_recurso and hora = @hora and dia = @dia

end

End Try

Begin Catch

 PRINT 'hubo error';

 SELECT ERROR_NUMBER() AS ErrorNumber,ERROR_MESSAGE() AS ErrorMessage;

```
RAISERROR ('HUBO ERROR', 16, 1);
```

```
End catch
```

```
End
```

```
Go
```

```
Insert into RyS_RecursosClub values ('Restaurante', 2, 'Edificio Principal', 'Alimentación')
```

```
Insert into RyS_RecursosClub values ('Bar', 2, 'Área Recreativa', 'Alimentación')
```

```
Insert into RyS_RecursosClub values ('Oficina', 2, 'Área Administrativa', 'Trabajo')
```

```
Insert into RyS_RecursosClub values ('Lobby', 2, 'Edificio Principal', 'Trabajo')
```

```
Insert into RyS_RecursosClub values ('Bar', 4, 'Área Principal', 'Alimentación')
```

```
Insert into RyS_RecursosClub values ('Piscina Grande', 4, 'Área Principal', 'Recreación')
```

```
Insert into RyS_RecursosClub values ('Piscina de niños', 4, 'Área Principal', 'Recreación')
```

```
Insert into RyS_RecursosClub values ('Parque acuático', 5, 'Área Principal', 'Recreación')
```

```
Insert into RyS_RecursosClub values ('Restaurante', 5, 'Edificio del lobby', 'Alimentación')
```

```
Insert into RyS_RecursosClub values ('Cancha de fútbol', 2, 'Área Recreativa', 'Recreación')
```

```
Insert into RyS_RecursosClub values ('Gimnasio', 2, 'Instalación de gimnasio', 'Recreación')
```

```
INSERT INTO RyS_Empleado VALUES ('Juan', 'Pérez', '9:00am-5:00pm', 2000, 2500, 'Asistente  
Administrativo', 'Activo', 10);
```

```
INSERT INTO RyS_Empleado VALUES ('Luis', 'Torres', 'L-V:10:00am-6:00pm', 1800, 2400,  
'Recepcionista', 'Activo', 11);
```

```
INSERT INTO RyS_Empleado VALUES ('María', 'González', 'L-V:8:00am-4:00pm', 2500, 3000, 'Chef',  
'Activo', 3);
```

```
INSERT INTO RyS_Empleado VALUES ('Rosa', 'Vargas', 'S-D:6:00am-2:00pm', 2000, 2500, 'Barista',  
'Activo', 3);
```

```
INSERT INTO RyS_Empleado VALUES ('Pedro', 'Ramírez', 'S-D:9:00am-5:00pm', 2500, 3000,  
'Salvavidas', 'Activo', 1);
```

INSERT INTO RyS_Empleado VALUES('Carolina', 'Sánchez', 'L-V:9:00am-5:00pm', 4500, 5000, 'Gerente', 'Activo', 10);

INSERT INTO RyS_Empleado VALUES('Luis', 'Sánchez', 'L-V:9:00am-5:00pm', 1500, 2000, 'Entrenador', 'Activo', 13);

INSERT INTO RyS_VacacionesEmpl VALUES ('2023-08-15', '2023-08-25', 3);

INSERT INTO RyS_VacacionesEmpl VALUES('2023-09-05', '2023-09-12', 3);

INSERT INTO RyS_VacacionesEmpl VALUES('2024-02-10', '2024-02-20', 6);

INSERT INTO RyS_VacacionesEmpl VALUES('2023-10-01', '2023-10-07', 7);

INSERT INTO RyS_VacacionesEmpl VALUES('2023-11-15', '2023-11-15', 9);

INSERT INTO RyS_VacacionesEmpl VALUES('2023-12-24', '2024-01-02', 10);

INSERT INTO RyS_VacacionesEmpl VALUES('2023/06/30', '2023/07/15', 11);

INSERT INTO RyS_VacacionesEmpl VALUES('2023/06/30', '2023/07/21', 12);

INSERT INTO RyS_VacacionesEmpl VALUES('2023/06/30', '2023/07/07', 4);

Insert into RyS_PagoHorasExtra values (2,2,'2023/06/22',3000)

Insert into RyS_PagoHorasExtra values (3,3,'2023/06/23',6750)

Insert into RyS_PagoHorasExtra values (4,1,'2023/06/24',4500)

Insert into RyS_PagoHorasExtra values (9,1,'2023/06/25',3000)

Insert into RyS_PagoHorasExtra values (10,2,'2023/06/26',5000)

Insert into RyS_Socio_Club values (2,1)

Insert into RyS_Socio_Club values (2,2)

Insert into RyS_Socio_Club values (2,9)

Insert into RyS_Socio_Club values (2,10)

Insert into RyS_Reserva_Espacio values ('Uso personal',1,1,'2023/06/23','9:00','12:00')

Insert into RyS_Reserva_Espacio values ('Uso personal',1,9,'2023/06/23','9:00','12:00')

Insert into RyS_Reserva_Espacio values ('Uso personal',1,10,'2023/06/23','9:00','12:00')

Insert into RyS_Mantenimiento values (2,2,'2023/06/23','Salon','5:00')

Insert into RyS_Eventos values ('Concierto de Toledo',2,'Se presentara Toledo con DJP ','Publico',
'2023/07/03', 1000)

Insert into RyS_Eventos values ('Feria de pymes',2,'Feria de pymes invitadas con sorpresas
, 'Publico', '2023/07/09',3000)

Insert into RyS_Eventos values ('Concierto Privado de Chayanne',2,'Concierto exclusivo de
Chayanne para socios','Privado', '2023/08/11', 1000)

Insert into RyS_Eventos values ('Campeonato relampago de futbol',2,'Pequeño campeonato de
futbol para socios','Privado', '2023/07/15', 1000)

Insert into RyS_ServiciosAprovechados values (13,2,13,'2023/06/29','9:00')

Insert into RyS_ServiciosAprovechados values (13,9,13,'2023/06/29','9:00')

Insert into RyS_ServiciosAprovechados values (13,1,13,'2023/06/29','9:00')

use IF5100_2023_Proyecto_RaSe

--Inserts dbo.RyS_Club

INSERT INTO dbo.RyS_Club(pais, recursos, ubicacion, id_gerente)

VALUES

('Estados Unidos', 20, 'Nueva York', 101),

('España', 15, 'Barcelona', 102),

('Francia', 14, 'París', 103),

('Brasil', 13, 'Río de Janeiro', 104),

('Australia', 19, 'Sídney', 105),

('Alemania', 5, 'Berlín', 6),
('Italia', 6, 'Roma', 7),
('México', 5, 'Ciudad de México', 8),
('Canadá', 7, 'Toronto', 9),
('Japón', 8, 'Tokio', 10);

--Inserts RyS_SocioDatosI

```
INSERT INTO RyS_SocioDatosI (nombre, apellido, cedula, plan_ofrece)
VALUES('Raúl', 'Miranda', '305460445', 'Básico'),
('Fernando', 'Fernandez', '609890986', 'Completo'),
('María', 'López', '0987654321', 'Estandar'),
('Juan', 'Pérez', '1234567890', 'Premiun'),
('Juan', 'Pérez', '123456789', 'Básico'),
('María', 'González', '234567890', 'Completo'),
('Pedro', 'Rodríguez', '345678901', 'Básico'),
('Laura', 'López', '456789012', 'Completo'),
('Carlos', 'Martínez', '567890123', 'Básico'),
('Ana', 'Hernández', '678901234', 'Completo');
```

--Inserts dbo.RyS_Club

```
INSERT INTO dbo.RyS_Club(pais, recursos, ubicacion, id_gerente)
VALUES
('Estados Unidos', 20, 'Nueva York', 101),
('España', 15, 'Barcelona', 102),
('Francia', 14, 'París', 103),
('Brasil', 13, 'Río de Janeiro', 104),
('Australia', 19, 'Sídney', 105),
('Alemania', 5, 'Berlín', 6),
```

```
('Italia', 6, 'Roma', 7),  
('México', 5, 'Ciudad de México', 8),  
('Canadá', 7, 'Toronto', 9),  
('Japón', 8, 'Tokio', 10);
```

```
--Inserts RyS_Contabilidad
```

```
INSERT INTO RyS_Contabilidad (monto, fecha_transaccion, id_club, tipo_pago, descrip_pago)  
VALUES (100, '2023-06-01', 1, 'Crédito', 'Pago mensual'),  
(50, '2023-06-02', 1, 'Efectivo', 'Pago por reserva'),  
(75, '2023-06-03', 1, 'Efectivo', 'Pago de membresía'),  
(120, '2023-06-04', 1, 'Débito', 'Pago de servicios adicionales'),  
(90, '2023-06-05', 1, 'Cheque', 'Pago de cuota mensual'),  
(80, '2023-06-06', 1, 'Crédito', 'Pago de clases de yoga'),  
(60, '2023-06-07', 1, 'Efectivo', 'Pago de cena en el restaurante'),  
(110, '2023-06-08', 1, 'Crédito', 'Pago de eventos'),  
(70, '2023-06-09', 1, 'Débito', 'Pago de membresía anual'),  
(95, '2023-06-10', 1, 'Cheque', 'Pago de alquiler de cancha');
```

```
--Inserts RyS_ContabilidadClub
```

```
INSERT INTO RyS_ContabilidadClub (id_pago, id_socio, estado)  
VALUES (14, 1, 'Pagado'),  
(5, 2, 'Cancelado'),  
(6, 1, 'Pendiente'),  
(7, 9, 'Cancelado'),  
(8, 10, 'Cancelado'),  
(9, 1, 'Pendiente'),  
(10, 2, 'Cancelado'),
```

```
(11, 21, 'Cancelado'),  
(12, 22, 'Cancelado'),  
(13, 22, 'Cancelado');
```

```
--Inserts RyS_Promociones
```

```
INSERT INTO RyS_Promociones (descripcion, inicio_promo, fin_promo, tipo, id_club)  
VALUES ('Promoción de verano', '2023-06-01', '2023-06-30', 'Descuento', 2),  
('Promoción de cumpleaños', '2023-07-01', '2023-07-31', 'Beneficio', 2),  
('Promoción de invierno', '2023-08-01', '2023-08-31', 'Descuento', 2),  
('Promoción de membresía', '2023-09-01', '2023-09-30', 'Beneficio', 2),  
('Promoción de otoño', '2023-10-01', '2023-10-31', 'Descuento', 2),  
('Promoción de aniversario', '2023-11-01', '2023-11-30', 'Beneficio', 2),  
('Promoción de primavera', '2023-12-01', '2023-12-31', 'Descuento', 2),  
('Promoción de renovación', '2024-01-01', '2024-01-31', 'Beneficio', 2),  
('Promoción de fin de año', '2024-02-01', '2024-02-28', 'Descuento', 2),  
('Promoción de bienvenida', '2024-03-01', '2024-03-31', 'Beneficio', 2);
```

```
--Inserts RyS_ContabilidadEventoPublico
```

```
INSERT INTO RyS_ContabilidadEventoPublico (id_pago, cod_entrada)  
VALUES (5, 001),  
(6, 002),  
(7, 003),  
(8, 004),  
(9, 005),  
(10, 006),  
(11, 007),
```

(12, 008),

(13, 009),

(14, 010);

--Insert RyS_Contrato

INSERT INTO RyS_Contrato (nombre_empresa, tipo_contrato, servicio, inicio_contrato, fin_contrato, id_club)

VALUES ('Empresa A', 'Servicio de limpieza', 'Contrato mensual', '2023-06-01', '2023-07-01', 2),

('Empresa B', 'Servicio de catering', 'Contrato anual', '2023-06-01', '2024-06-01', 2),

('Empresa C', 'Servicio de mantenimiento', 'Contrato mensual', '2023-06-01', '2023-07-01', 2),

('Empresa D', 'Servicio de seguridad', 'Contrato anual', '2023-06-01', '2024-06-01', 2),

('Empresa E', 'Servicio de jardinería', 'Contrato mensual', '2023-06-01', '2023-07-01', 2),

('Empresa F', 'Servicio de limpieza', 'Contrato anual', '2023-06-01', '2024-06-01', 2),

('Empresa G', 'Servicio de catering', 'Contrato mensual', '2023-06-01', '2023-07-01', 2),

('Empresa H', 'Servicio de mantenimiento', 'Contrato anual', '2023-06-01', '2024-06-01', 2),

('Empresa I', 'Servicio de seguridad', 'Contrato mensual', '2023-06-01', '2023-07-01', 2),

('Empresa J', 'Servicio de jardinería', 'Contrato anual', '2023-06-01', '2024-06-01', 2);

--Insert RyS_ServiciosAprovechados

INSERT INTO RyS_ServiciosAprovechados (id_recurso, id_socio, id_empleado, fechas, horas)

VALUES (1, 1, 2, '2023-06-01', '10:59'),

(2, 2, 3, '2023-06-02', '14:30'),

(3, 9, 4, '2023-06-03', '17:00'),

(4, 10, 6, '2023-06-04', '09:00'),

(5, 18, 7, '2023-06-05', '12:30'),

(6, 19, 8, '2023-06-06', '16:00'),

(7, 20, 9, '2023-06-07', '11:00'),


```
(8, 21, 10, '2023-06-08', '15:30'),  
(9, 22, 11, '2023-06-09', '13:00'),  
(10, 23, 12, '2023-06-10', '18:00');
```

```
--Insert RyS_SocioDatosExtra
```

```
INSERT INTO RyS_SocioDatosExtra (id_socio, correo, fecha_afiliacion)  
VALUES (1, 'raul.miranda@ucr.ac.cr', '2023-06-21'),  
(2, 'fernando.fernandez@ucr.ac.cr', '2023-06-21'),  
(9, 'juanperez12@gmail.com', '2023-03-01'),  
(10, 'maria.l@gmail.com', '2023-04-01'),  
(18, 'juan@gmail.com', '2023-05-01'),  
(19, 'mariag@gmail.com', '2023-06-01'),  
(20, 'p.gonzalez@gmail.com', '2023-07-01'),  
(21, 'lauralopez@gmail.com', '2023-08-01'),  
(22, 'carlos.martinez@gmail.com', '2023-09-01'),  
(23, 'ana@gmail.com', '2023-10-01');
```

```
--Insert RyS_Socio_Club
```

```
INSERT INTO RyS_Socio_Club (id_club, id_socio)  
VALUES (2, 1),  
(2, 2),  
(2, 9),  
(2, 10),  
(2, 18),  
(2, 19),  
(2, 20),  
(2, 21),
```

(2, 22),

(2, 23);

--Inserts RyS_SocioProm

INSERT INTO RyS_SocioProm (id_promo, id_socio)

VALUES

(2, 1),

(3, 2),

(4, 9),

(5, 10),

(6, 18),

(7, 19),

(8, 20),

(9, 21),

(10, 22),

(11, 23);

--Insert RyS_Invitado

INSERT INTO RyS_Invitado (nombre, apellido, cedula, id_socio)

VALUES ('Nicanor', 'Beltrán', '123456789', 1),

('Zafiro', 'Montenegro', '234567890', 2),

('Quintín', 'Sandoval', '345678901', 9),

('Lirio', 'Navarro', '456789012', 10),

('Eleuteria', 'Palma', '567890123', 18),

('Celestino', 'Flores', '678901234', 19),

('Odalys', 'Cortés', '789012345', 20),

('Filiberto', 'Vega', '890123456', 21),

('Luminita', 'López', '901234567', 22),

```
('Evarista', 'Gómez', '012345678', 23);
```

```
--Insert RyS_Promociones
```

```
INSERT INTO RyS_Promociones (descripcion, inicio_promo, fin_promo, tipo, id_club)
```

```
VALUES
```

```
('Promoción de cumpleaños', '2023-07-01', '2023-07-31', 'Beneficio', 2),
```

```
('Promoción de invierno', '2023-08-01', '2023-08-31', 'Descuento', 2),
```

```
('Promoción de membresía', '2023-09-01', '2023-09-30', 'Beneficio', 2),
```

```
('Promoción de otoño', '2023-10-01', '2023-10-31', 'Descuento', 2),
```

```
('Promoción de aniversario', '2023-11-01', '2023-11-30', 'Beneficio', 2),
```

```
('Promoción de primavera', '2023-12-01', '2023-12-31', 'Descuento', 2),
```

```
('Promoción de renovación', '2024-01-01', '2024-01-31', 'Beneficio', 2),
```

```
('Promoción de fin de año', '2024-02-01', '2024-02-28', 'Descuento', 2),
```

```
('Promoción de bienvenida', '2024-03-01', '2024-03-31', 'Beneficio', 2),
```

```
('Promoción de verano', '2023-06-01', '2023-06-30', 'Descuento', 2);
```

```
Select * from RyS_Personal_Evento
```

```
Select * from RyS_Eventos
```

```
Select * from RyS_Empleado
```

```
Select * from RyS_RecursosClub
```

```
Insert into RyS_Evento_Privado values (1,9,4),(2,11,2),(9,11,5),(18,11,3),(21,11,2),(22,11,3)
```

Insert into RyS_Evento_Publico values

(8,'A2',5000),(8,'A3',5000),(8,'A4',5000),(8,'B1',3000),(8,'B2',3000),(8,'B3',3000),(8,'C2',2000),(8,'C3',2000)

Insert into RyS_Personal_Evento values (11,1,'Seguridad'),(10,1,'Bebidas'),(3,1,'Seguridad')