

```
-- Comando para borrar tabla ya creada //////////DROP Table PLANES
/*Esta es la secuencia para crear la table en comando SQL*/
-- Creación de la tabla Planes
CREATE TABLE Planes (
    ID INT PRIMARY KEY,
    Nombre VARCHAR2(255),
    Tipo CHAR(1) CHECK (Tipo IN ('A', 'E')),

    MatriculaPersona DECIMAL(7, 2),
    CuotaMensual DECIMAL(7, 2),
    Descripcion VARCHAR2(1000)
);

-- Creación de la tabla Socios para almacenar datos comunes de
CREATE TABLE Socios (
    ID_Socio INT PRIMARY KEY,
    Nombre VARCHAR2(255),
    Direccion VARCHAR2(255),
    Telefono VARCHAR2(15),
    Email VARCHAR2(100),
    ID_Plan INT,
    FechaInscripcion DATE,
    FechaCaducidad DATE,
    CONSTRAINT fk_plan FOREIGN KEY (ID_Plan) REFERENCES Planes(ID)
);

-- Creación de la tabla TrabajadoresCorporativos para aquellos
CREATE TABLE TrabajadoresCorporativos (
    ID_Socio INT PRIMARY KEY,
    NIF_Empresa VARCHAR2(9),
    CONSTRAINT fk_socio_corporativo FOREIGN KEY (ID_Socio) REFER
```

```
-- Creación de la tabla TrabajadoresAutonomos para los responsa
CREATE TABLE TrabajadoresAutonomos (
    ID_Socio INT PRIMARY KEY,
    Banco VARCHAR2(255),
    NumeroCuenta VARCHAR2(24),
    CONSTRAINT fk_socio_autonomo FOREIGN KEY (ID_Socio) REFERENC
);

/*Una vez creadas las tablas, solo he encontrado dos soluciones
ALTER TABLE Socios ADD DocumentoIdentidad VARCHAR2(20);
ALTER TABLE Socios ADD Apellido1 VARCHAR2(255);
ALTER TABLE Socios ADD Apellido2 VARCHAR2(255);
ALTER TABLE Socios ADD Sexo CHAR(1) CHECK (Sexo IN ('M', 'F', 'O'));
ALTER TABLE Socios ADD FechaNacimiento DATE;
ALTER TABLE Socios ADD FechaAlta DATE;
ALTER TABLE Socios ADD NumeroTarjetaAcceso VARCHAR2(20);
ALTER TABLE Socios ADD Ciudad VARCHAR2(255);
ALTER TABLE Socios ADD CodigoPostal VARCHAR(10);
ALTER TABLE Socios ADD Observaciones VARCHAR2(1000);
ALTER TABLE Socios ADD Activo CHAR(1) check (Activo IN('S', 'N'));

CREATE TABLE Empresas (
    NIF VARCHAR2(9) PRIMARY KEY,
    NombreEmpresa VARCHAR2(255),
    TelefonoContacto VARCHAR2(15),
    PersonaContacto VARCHAR2(255),
    EmailContacto VARCHAR2(100),
    FechaInicioConvenio DATE,
    FechaFinConvenio DATE
);

CREATE TABLE HistoricoPlanesSocios (
    ID_Historico INT PRIMARY KEY,
    ID_Socio INT NOT NULL,
    FechaCambio DATE NOT NULL,
    PlanAnterior INT,
    PlanActual INT,
```

```
BajaDefinitiva CHAR(1) CHECK (BajaDefinitiva IN ('S', 'N')),
CONSTRAINT fk_historico_socio FOREIGN KEY (ID_Socio) REFEREN
CONSTRAINT fk_plan_anterior FOREIGN KEY (PlanAnterior) REFER
CONSTRAINT fk_plan_actual FOREIGN KEY (PlanActual) REFERENCE
);
/*En esta seccion hacemos un disparador para que la tabla histor
ID_Plan seda de baja definitivamente con el valor NULL*/
CREATE OR REPLACE TRIGGER trg_after_plan_drop
AFTER UPDATE OF ID_Plan ON Socios
FOR EACH ROW
WHEN (OLD.ID_Plan IS NOT NULL AND NEW.ID_Plan IS NULL)
BEGIN
    INSERT INTO HistoricoPlanesSocios (
        ID_Historico, ID_Socio, FechaCambio, PlanAnterior, PlanA
    ) VALUES (
        historico_seq.NEXTVAL, :OLD.ID_Socio, SYSDATE, :OLD.ID_P
    );
END;
/*Secuencia para crear el ID_Socio automaticamente por sistema.
CREATE SEQUENCE es para crear la secuencia con el nombre socio_
START WITH 1 indica que la secuencia comienza siempre con el va
CREATE SEQUENCE socio_seq START WITH 1;
CREATE OR REPLACE TRIGGER Socios_before_insert
BEFORE INSERT ON Socios
FOR EACH ROW
BEGIN
    SELECT socio_seq.NEXTVAL INTO :new.ID_Socio FROM dual;
END;

CREATE SEQUENCE historico_seq START WITH 1;

/*Esta seccion es un disparador para que la tabla de historial
cada vez que hay cambion en el ID_PLAN de la tabla de socios*/
CREATE OR REPLACE TRIGGER trg_after_plan_change
AFTER UPDATE OF ID_Plan ON Socios
```

```
WHEN (OLD.ID_Plan IS NOT NULL AND NEW.ID_Plan IS NOT NULL AND OLD.ID_Plan < NEW.ID_Plan)
BEGIN
    INSERT INTO HistoricoPlanesSocios (
        ID_Historico, ID_Socio, FechaCambio, PlanAnterior, PlanActual
    ) VALUES (
        historico_seq.NEXTVAL, :OLD.ID_Socio, SYSDATE, :OLD.ID_Plan, :NEW.ID_Plan
    );
END;
```