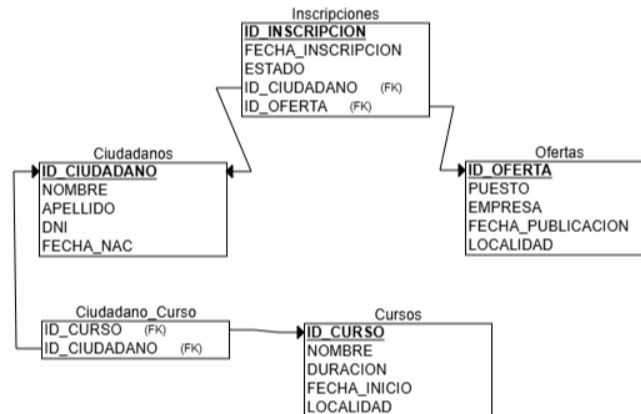


Vamos a crear una base de datos para gestionar usuarios del servicio público de empleo y ofertas a las que se inscriben. Comenzaremos creando una base de datos llamada SEPE y después emplearemos la información dada.



1. CREACIÓN DE LA BASE DE DATOS Y TABLAS

Utilizar las siguientes instrucciones para crear las tablas de la base de datos

```
CREATE TABLE Ciudadanos (  
    ID_CIUDADANO INT PRIMARY KEY,  
    NOMBRE VARCHAR(50),  
    APELLIDO VARCHAR(50),  
    DNI VARCHAR(10) UNIQUE,  
    FECHA_NAC DATE  
);  
  
CREATE TABLE Ofertas (  
    ID_OFERTA INT PRIMARY KEY,  
    PUESTO VARCHAR(100),  
    EMPRESA VARCHAR(100),  
    FECHA_PUBLICACION DATE,  
    LOCALIDAD VARCHAR(50)  
);  
  
CREATE TABLE Inscripciones (  
    ID_INSCRIPCION INT PRIMARY KEY,  
    FECHA_INSCRIPCION DATE,  
    ESTADO VARCHAR(20),  
    ID_CIUDADANO INT FOREIGN KEY REFERENCES Ciudadanos(ID_CIUDADANO),  
    ID_OFERTA INT FOREIGN KEY REFERENCES Ofertas(ID_OFERTA)  
);  
  
CREATE TABLE Cursos (  
    ID_CURSO INT PRIMARY KEY,  
    NOMBRE VARCHAR(100),  
    DURACION INT,  
    FECHA_INICIO DATE,  
    LOCALIDAD VARCHAR(50)  
);  
  
CREATE TABLE Ciudadano_Curso (  
    ID_CURSO INT FOREIGN KEY REFERENCES Cursos(ID_CURSO),  
    ID_CIUDADANO INT FOREIGN KEY REFERENCES Ciudadanos(ID_CIUDADANO)  
);
```

```
        ID_OFERTA INT FOREIGN KEY REFERENCES Ofertas(ID_OFERTA),
        FECHA_INSCRIPCION DATE,
        ESTADO VARCHAR(20) CHECK (ESTADO IN ('Pendiente', 'Seleccionado',
'Rechazado'))
    );

CREATE TABLE Ciudadano_Curso (
    ID_CIUDADANOINSCRITO INT,
    ID_CURSOINSCRITO INT,
)

CREATE TABLE Cursos (
    ID_CURSO INT PRIMARY KEY,
    NOMBRE VARCHAR(100),
    DURACION INT, -- en horas
    FECHA_INICIO DATE,
    LOCALIDAD VARCHAR(50)
);
```

2. INSERCIÓN DE 5 REGISTROS POR TABLA

```
INSERT INTO Ciudadanos (ID_CIUDADANO, NOMBRE, APELLIDO, DNI,
FECHA_NAC) VALUES
(1,'Laura', 'Lozano', '12345678A', '1985-04-10'),
(2,'Pedro', 'López', '87654321B', '1990-06-22'),
(3,'Sofía', 'Ruiz', '11223344C', '1995-01-15'),
(4,'Javier', 'Sánchez', '55667788D', '1978-12-01'),
(5,'Marta', 'González', '99887766E', '1988-07-09');

INSERT INTO Ofertas (ID_OFERTA, PUESTO, EMPRESA, FECHA_PUBLICACION,
LOCALIDAD) VALUES
(1,'Administrativo', 'Ofitec', '2024-01-10', 'Madrid'),
(2,'Técnico de Redes', 'NetSolutions', '2024-02-20', 'Barcelona'),
(3,'Desarrollador Web', 'CodeFactory', '2024-03-01', 'Valencia'),
(4,'Mozo de almacén', 'LogiTrans', '2024-04-05', 'Sevilla'),
(5,'Analista de datos', 'DataCorp', '2024-04-15', 'Bilbao');
```

```
INSERT INTO Inscripciones (ID_INSCRIPCION, ID_CIUDADANO, ID_OFERTA,
FECHA_INSCRIPCION, ESTADO) VALUES
(1, 1, 1, '2024-01-15', 'Pendiente'),
(2, 2, 2, '2024-02-22', 'Seleccionado'),
(3, 3, 3, '2024-03-05', 'Pendiente'),
(4, 4, 4, '2024-04-07', 'Rechazado'),
(5, 5, 5, '2024-04-18', 'Pendiente');

INSERT INTO Cursos (ID_CURSO, NOMBRE, DURACION, FECHA_INICIO,
LOCALIDAD) VALUES
(1, 'Ofimática básica', 40, '2024-05-01', 'Madrid'),
(2, 'Redes Cisco', 60, '2024-05-10', 'Barcelona'),
(3, 'Desarrollo Frontend', 80, '2024-06-01', 'Valencia'),
(4, 'Logística y almacén', 30, '2024-06-15', 'Sevilla'),
(5, 'Big Data y análisis', 90, '2024-07-01', 'Bilbao');
```

3. Ejercicios de funciones

1. Devolver el número de años de experiencia desde la fecha de inscripción. La función recibe el id de la inscripción

```
CREATE FUNCTION AniosInscrito(@ID_INSCRIPCION INT)
RETURNS INT
AS
BEGIN
    DECLARE @anios INT;

    SELECT @anios = DATEDIFF(YEAR, FECHA_INSCRIPCION, GETDATE())
    FROM Inscripciones WHERE ID_INSCRIPCION = @ID_INSCRIPCION;

    RETURN @anios;
END;
```

2. Obtener nombre completo del ciudadano. La función recibe el id del ciudadano

```
CREATE FUNCTION NombreCompleto(@ID_CIUDADANO INT)
RETURNS VARCHAR(100)
AS
BEGIN
```

```
DECLARE @nombreCompleto VARCHAR(100);

SELECT @nombreCompleto = NOMBRE + ' ' + APELLIDO FROM Ciudadanos
WHERE ID_CIUDADANO = @ID_CIUDADANO;

RETURN @nombreCompleto;

END;
```

3. Listar ciudadanos inscritos en una localidad dada

```
CREATE FUNCTION CiudadanosPorLocalidad(@Localidad VARCHAR(50))
RETURNS TABLE
AS
RETURN (
    SELECT Ciudadanos.*
    FROM Ciudadanos
    JOIN Inscripciones ON Ciudadanos.ID_CIUDADANO =
Inscripciones.ID_CIUDADANO
    JOIN Ofertas ON Inscripciones.ID_OFERTA = Ofertas.ID_OFERTA
    WHERE Ofertas.LOCALIDAD = @Localidad
);
```

4. Devolver estado según fecha de publicación de la oferta. La función recibe el id de la oferta. Si se ha publicado hace menos de 30 días devolverá la palabra: actual. Si se ha publicado hace entre 30 y 60 días es reciente y si se ha publicado hace más de 60 es antigua

```
CREATE FUNCTION EstadoOferta(@ID_OFERTA INT)
RETURNS VARCHAR(50)
AS
BEGIN
    DECLARE @estado VARCHAR(50);
    DECLARE @fecha DATE;

    SELECT @fecha = FECHA_PUBLICACION FROM Ofertas WHERE ID_OFERTA =
@ID_OFERTA;

    SELECT @estado = CASE
        WHEN DATEDIFF(DAY, @fecha, GETDATE()) < 30 THEN 'actual'
        WHEN DATEDIFF(DAY, @fecha, GETDATE()) >= 30 AND DATEDIFF(DAY,
@fecha, GETDATE()) <= 60 THEN 'Reciente'
        ELSE 'Antigua'
    END;

    END;
```

```
        RETURN @estado;  
END;
```

5. Función que devuelva inscripciones pendientes por oferta. La función recibe el id de oferta y devuelve los datos de las inscripciones en estado 'pendiente'

```
CREATE FUNCTION PendientesPorOferta(@ID_OFERTA INT)  
RETURNS TABLE  
AS  
RETURN (  
    SELECT * FROM Inscripciones  
    WHERE ID_OFERTA = @ID_OFERTA AND ESTADO = 'Pendiente'  
);
```

6. Función que devuelva ciudadanos con más de una inscripción

```
CREATE FUNCTION CiudadanosConMultiplesInscripciones()  
RETURNS TABLE  
AS  
RETURN (  
    SELECT ID_CIUDADANO, COUNT(*) AS NumInscripciones  
    FROM Inscripciones  
    GROUP BY ID_CIUDADANO  
    HAVING COUNT(*) > 1  
);
```