

### Pregunta 1 (2 p.)

- Cree las sentencias que permitan la creación del modelo de datos presentados, incluyendo los tipos de datos, las llaves primarias y las llaves foráneas.

```
CREATE TABLE Asociado (  
    codigo_asociado int NOT NULL,  
    apellido_paterno varchar(30) NOT NULL,  
    apellido_materno varchar(30) NOT NULL,  
    nombre varchar(30) NOT NULL,  
    tipo_asociado char(1) NOT NULL,  
    numeroDocumento varchar(11) NOT NULL,  
    fechaUltimaCap date NOT NULL,  
    telefonoFijo varchar(7) NOT NULL,  
    telefonoTrabajo varchar(7) NOT NULL,  
    telefonoMovil varchar(9) NOT NULL,  
    Distrito_codigo_distrito int NOT NULL,  
    Especialidad_codigo_especialidad int NOT NULL,  
    Centro_Ocupacional_codigo_co int NOT NULL,  
    CONSTRAINT Asociado_pk PRIMARY KEY (codigo_asociado)  
);  
  
-- Table: Centro_Ocupacional  
CREATE TABLE Centro_Ocupacional (  
    codigo_co int NOT NULL,  
    razon_social varchar(30) NOT NULL,  
    telefonoFijo varchar(7) NOT NULL,  
    Distrito_codigo_distrito int NOT NULL,  
    Giro_codigo_giro int NOT NULL,  
    CONSTRAINT Centro_Ocupacional_pk PRIMARY KEY (codigo_co)  
);  
  
-- Table: Curso  
CREATE TABLE Curso (  
    codigo_curso int NOT NULL,  
    descripcion_curso varchar(30) NOT NULL,  
    nro_HorasT int NOT NULL,  
    nro_HorasP int NOT NULL,  
    nivelComplejidad char(1) NOT NULL,  
    CONSTRAINT Curso_pk PRIMARY KEY (codigo_curso)  
);  
  
-- Table: Distrito  
CREATE TABLE Distrito (  
    codigo_distrito int NOT NULL,  
    nombre_distrito varchar(30) NOT NULL,  
    CONSTRAINT Distrito_pk PRIMARY KEY (codigo_distrito)  
);  
  
-- Table: Especialidad  
CREATE TABLE Especialidad (  
    codigo_especialidad int NOT NULL,  
    descripcion_especialidad varchar(30) NOT NULL,  
    CONSTRAINT Especialidad_pk PRIMARY KEY (codigo_especialidad)  
);  
  
-- Table: Evento  
CREATE TABLE Evento (  
    numero_evento int NOT NULL,  
    fecha_inicio date NOT NULL,  
    fecha_final date NOT NULL,  
    Curso_codigo_curso int NOT NULL,
```

```

Centro_Ocupacional_codigo_co int NOT NULL,
Expositor_codigo_expositor int NOT NULL,
frecuencia char(1) NOT NULL,
turno char(1) NOT NULL,
presupuesto decimal(10,2) NOT NULL,
CONSTRAINT Evento_pk PRIMARY KEY (numero_evento)
);

-- Table: Expositor
CREATE TABLE Expositor (
    codigo_expositor int NOT NULL,
    apellido_paterno varchar(30) NOT NULL,
    apellido_materno varchar(30) NOT NULL,
    nombres varchar(30) NOT NULL,
    telefonoMovil varchar(9) NOT NULL,
    eMail varchar(80) NOT NULL,
    Distrito_codigo_distrito int NOT NULL,
    Centro_Ocupacional_codigo_co int NOT NULL,
    CONSTRAINT Expositor_pk PRIMARY KEY (codigo_expositor)
);

-- Table: Giro
CREATE TABLE Giro (
    codigo_giro int NOT NULL,
    nombre_giro varchar(30) NOT NULL,
    CONSTRAINT Giro_pk PRIMARY KEY (codigo_giro)
);

-- Table: Inscripcion
CREATE TABLE Inscripcion (
    Evento_numero_evento int NOT NULL,
    Asociado_codigo_asociado int NOT NULL,
    asistido bit NOT NULL,
    CONSTRAINT Inscripcion_pk PRIMARY KEY
(Evento_numero_evento,Asociado_codigo_asociado)
);

-- foreign keys
-- Reference: Asociado_Centro_Ocupacional (table: Asociado)
ALTER TABLE Asociado ADD CONSTRAINT Asociado_Centro_Ocupacional
FOREIGN KEY (Centro_Ocupacional_codigo_co)
REFERENCES Centro_Ocupacional (codigo_co);

-- Reference: Asociado_Distrito (table: Asociado)
ALTER TABLE Asociado ADD CONSTRAINT Asociado_Distrito
FOREIGN KEY (Distrito_codigo_distrito)
REFERENCES Distrito (codigo_distrito);

-- Reference: Asociado_Especialidad (table: Asociado)
ALTER TABLE Asociado ADD CONSTRAINT Asociado_Especialidad
FOREIGN KEY (Especialidad_codigo_especialidad)
REFERENCES Especialidad (codigo_especialidad);

-- Reference: Centro_Ocupacional_Distrito (table: Centro_Ocupacional)
ALTER TABLE Centro_Ocupacional ADD CONSTRAINT Centro_Ocupacional_Distrito
FOREIGN KEY (Distrito_codigo_distrito)
REFERENCES Distrito (codigo_distrito);

-- Reference: Centro_Ocupacional_Giro (table: Centro_Ocupacional)
ALTER TABLE Centro_Ocupacional ADD CONSTRAINT Centro_Ocupacional_Giro
FOREIGN KEY (Giro_codigo_giro)
REFERENCES Giro (codigo_giro);

```

```

-- Reference: Evento_Centro_Ocupacional (table: Evento)
ALTER TABLE Evento ADD CONSTRAINT Evento_Centro_Ocupacional
    FOREIGN KEY (Centro_Ocupacional_codigo_co)
    REFERENCES Centro_Ocupacional (codigo_co);

-- Reference: Evento_Curso (table: Evento)
ALTER TABLE Evento ADD CONSTRAINT Evento_Curso
    FOREIGN KEY (Curso_codigo_curso)
    REFERENCES Curso (codigo_curso);

-- Reference: Evento_Expositor (table: Evento)
ALTER TABLE Evento ADD CONSTRAINT Evento_Expositor
    FOREIGN KEY (Expositor_codigo_expositor)
    REFERENCES Expositor (codigo_expositor);

-- Reference: Expositor_Centro_Ocupacional (table: Expositor)
ALTER TABLE Expositor ADD CONSTRAINT Expositor_Centro_Ocupacional
    FOREIGN KEY (Centro_Ocupacional_codigo_co)
    REFERENCES Centro_Ocupacional (codigo_co);

-- Reference: Expositor_Distrito (table: Expositor)
ALTER TABLE Expositor ADD CONSTRAINT Expositor_Distrito
    FOREIGN KEY (Distrito_codigo_distrito)
    REFERENCES Distrito (codigo_distrito);

-- Reference: Inscripcion_Asociado (table: Inscripcion)
ALTER TABLE Inscripcion ADD CONSTRAINT Inscripcion_Asociado
    FOREIGN KEY (Asociado_codigo_asociado)
    REFERENCES Asociado (codigo_asociado);

-- Reference: Inscripcion_Evento (table: Inscripcion)
ALTER TABLE Inscripcion ADD CONSTRAINT Inscripcion_Evento
    FOREIGN KEY (Evento_numero_evento)
    REFERENCES Evento (numero_evento);

```

Pregunta 2 (2 p.)

- Cree las sentencias que permita insertar por lo menos cinco (5) registros en dos (2) de las tablas creadas en la pregunta 1.

```

INSERT INTO Curso (codigo_curso, descripcion_curso, nro_HorasT, nro_HorasP,
nivelComplejidad)

```

```

VALUES

```

```

(1, 'Curso Diseño de Paginas Web', 20, 10, 'B'),
(2, 'Curso Base de Datos', 15, 5, 'A'),
(3, 'Curso Cocina', 30, 20, 'C'),
(4, 'Curso Derecho Empresarial', 25, 15, 'B'),
(5, 'Curso Electricidad', 10, 8, 'A');

```

```

INSERT INTO Distrito (codigo_distrito, nombre_distrito)

```

```

VALUES

```

```

(1, 'Distrito VMT'),
(2, 'Distrito SJL'),
(3, 'Distrito VES'),
(4, 'Distrito ATE'),
(5, 'Distrito ANCON');

```

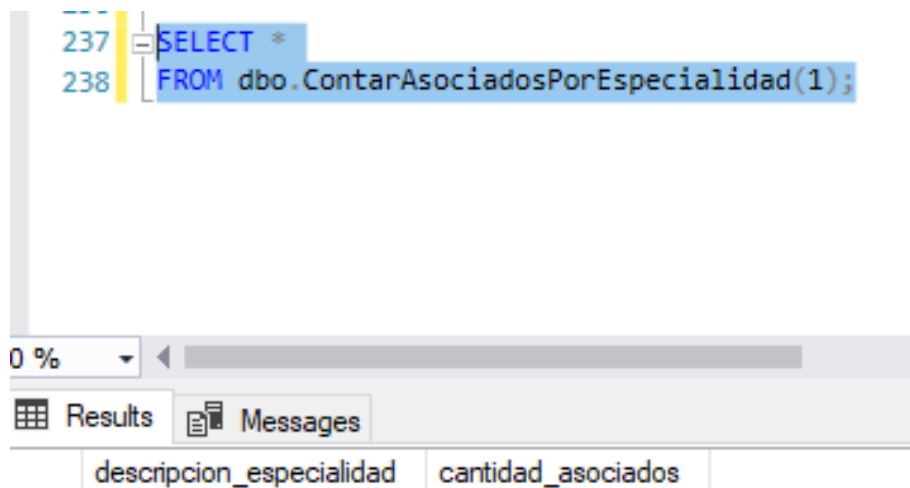
	codigo_curso	descripcion_curso	nro_HorasT	nro_HorasP	nivelComplejidad
1	1	Curso Diseño de Paginas Web	30	15	C
2	2	Curso Base de Datos	25	10	B
3	3	Curso Cocina	30	20	C
4	4	Curso Derecho Empresarial	35	18	B
5	5	Curso Electricidad	15	12	A

	codigo_distrito	nombre_distrito
1	1	Distrito VMT
2	2	Distrito SJL
3	3	Distrito VES
4	4	Distrito ATE
5	5	Distrito ANCON

### Pregunta 3 (2 p.)

□ Diseñar la función o procedimiento almacenado que permita determinar la cantidad de asociados por tipos de especialidad para un determinado centro ocupacional.

```
CREATE FUNCTION ContarAsociadosPorEspecialidad (
    @codigo_co INT
)
RETURNS TABLE
AS
RETURN (
    SELECT E.descripcion_especialidad, COUNT(A.codigo_asociado) AS
cantidad_asociados
    FROM Asociado A
    INNER JOIN Especialidad E ON A.Especialidad_codigo_especialidad =
E.codigo_especialidad
    WHERE A.Centro_Ocupacional_codigo_co = @codigo_co
    GROUP BY E.descripcion_especialidad
);
SELECT *
FROM dbo.ContarAsociadosPorEspecialidad(1)
```



#### Pregunta 4 (2 p.)

Diseñar el procedimiento almacenado que permita imprimir la cantidad de eventos y el monto total de presupuesto asignado para un rango de fechas determinado.

```
CREATE PROCEDURE ImprimirEventosPorFecha (  
    @fecha_inicio DATE,  
    @fecha_final DATE  
)  
AS  
BEGIN  
    SELECT COUNT(*) AS cantidad_eventos, SUM(presupuesto) AS total_presupuesto  
    FROM Evento  
    WHERE fecha_inicio >= @fecha_inicio AND fecha_final <= @fecha_final;  
END;  
  
EXEC ImprimirEventosPorFecha '2023-01-01', '2023-12-31'
```

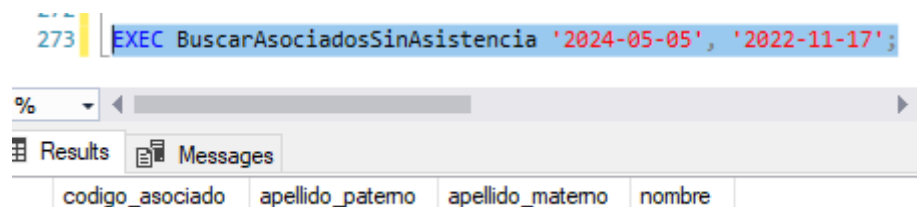


	cantidad_eventos	total_presupuesto
1	0	NULL

#### Pregunta 5 (4 p.)

☐ Diseñar la función o procedimiento almacenado que permita determinar el o (los) asociado(s) que no asistieron a ningún evento que se dio en un rango de fechas determinado.

```
CREATE PROCEDURE BuscarAsociadosSinAsistencia (  
    @fecha_inicio DATE,  
    @fecha_final DATE  
)  
AS  
BEGIN  
    SELECT A.codigo_asociado, A.apellido_paterno, A.apellido_materno, A.nombre  
    FROM Asociado A  
    WHERE NOT EXISTS (  
        SELECT 1  
        FROM Inscripcion I  
        INNER JOIN Evento E ON I.Evento_numero_evento = E.numero_evento  
        WHERE I.Asociado_codigo_asociado = A.codigo_asociado  
        AND E.fecha_inicio >= @fecha_inicio AND E.fecha_final <= @fecha_final  
    );  
END;
```

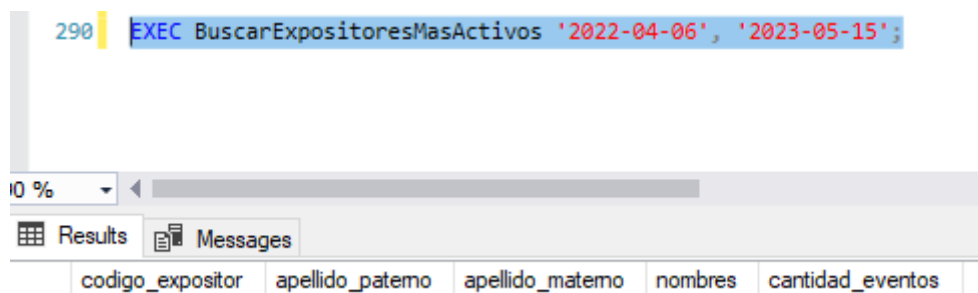


codigo_asociado	apellido_paterno	apellido_materno	nombre
-----------------	------------------	------------------	--------

#### Pregunta 6 (4 p.)

☐ Diseñar la función o procedimiento almacenado que permita determinar el(los) expositor(es) que más eventos han realizado en un rango de fechas determinado.

```
CREATE PROCEDURE BuscarExpositoresMasActivos (
    @fecha_inicio DATE,
    @fecha_final DATE
)
AS
BEGIN
    SELECT TOP 1 WITH TIES E.codigo_expositor, E.apellido_paterno,
E.apellido_materno, E.nombres, COUNT(*) AS cantidad_eventos
    FROM Evento EV
    INNER JOIN Expositor E ON EV.Expositor_codigo_expositor = E.codigo_expositor
    WHERE EV.fecha_inicio >= @fecha_inicio AND EV.fecha_final <= @fecha_final
    GROUP BY E.codigo_expositor, E.apellido_paterno, E.apellido_materno,
E.nombres
    ORDER BY COUNT(*) DESC;
END;
```



#### Pregunta 7 (4 p.)

☐ Diseñar la función o procedimiento almacenado que permita determinar el(los) curso(s) con la mayor cantidad de eventos en un rango de fechas determinado.

```
CREATE PROCEDURE BuscarCursosMasPopulares (
    @fecha_inicio DATE,
    @fecha_final DATE
)
AS
BEGIN
    SELECT TOP 1 WITH TIES C.codigo_curso, C.descripcion_curso, COUNT(*) AS
cantidad_eventos
    FROM Evento E
    INNER JOIN Curso C ON E.Curso_codigo_curso = C.codigo_curso
    WHERE E.fecha_inicio >= @fecha_inicio AND E.fecha_final <= @fecha_final
    GROUP BY C.codigo_curso, C.descripcion_curso
    ORDER BY COUNT(*) DESC;
END;

EXEC BuscarCursosMasPopulares '2021-06-07', '2023-08-05';
```

```
307 EXEC BuscarCursosMasPopulares '2021-06-07', '2023-08-05';
```

10 %



Results



Messages

codigo_curso	descripcion_curso	cantidad_eventos
--------------	-------------------	------------------

