



Bases de Datos SQL

Ejercicio resuelto “Pinacotecas”

Pinacotecas

El ministerio encargado de temas culturales de cierto país quiere mantener información acerca de todos los cuadros que se encuentran en sus pinacotecas y toda la información relacionada con ellos.

De cada pinacoteca se desea saber el nombre (que se supone único) , la ciudad en que se encuentra, la dirección y los metros cuadrados que tiene.

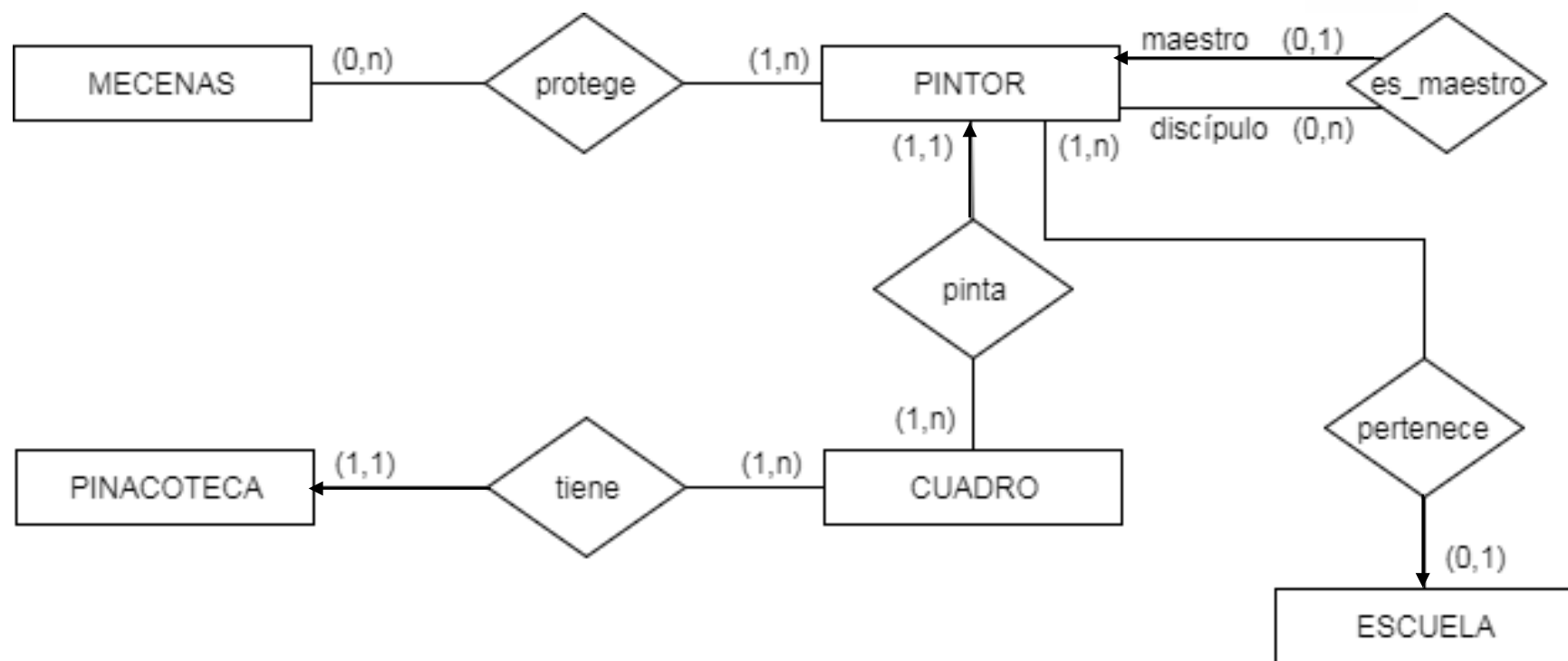
Cada pinacoteca tiene una serie de cuadros de los que quiere mantener información acerca de su código, nombre, dimensiones, fecha en que fue pintado y técnica utilizada.

Cada cuadro es pintado por un determinado pintor (nombre, país, ciudad, fecha de nacimiento y fecha de defunción si procede). Un pintor puede tener a otro como maestro; a su vez, un maestro puede serlo de varios (o de ninguno).

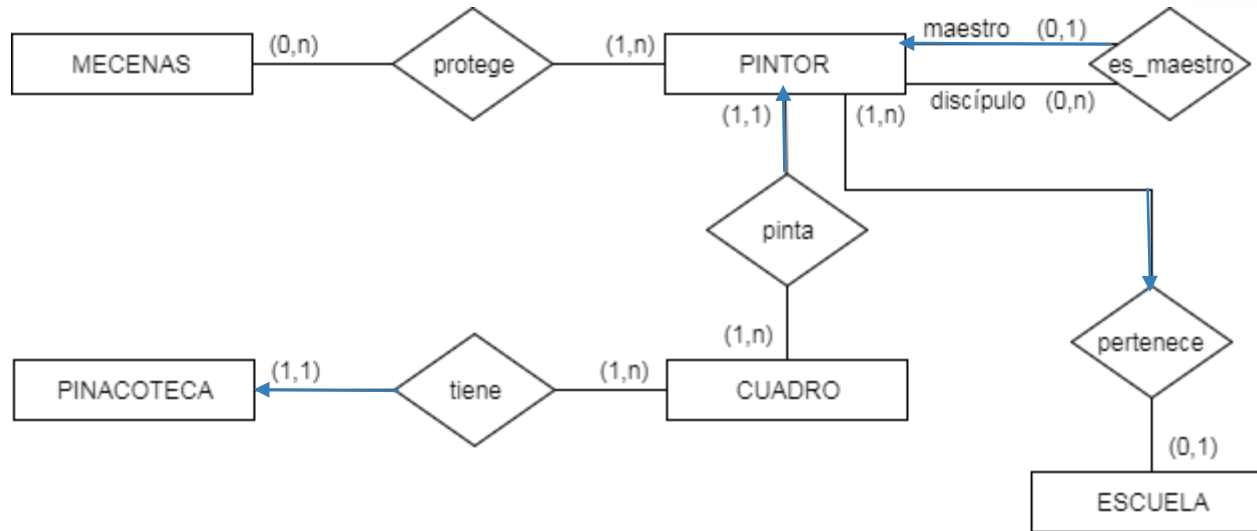
Los pintores pueden pertenecer o no a una escuela de la que se desea saber su nombre y en qué país y en qué fecha apareció.

Los pintores pueden tener también uno o varios mecenas que les protegen (nombre, país, fecha de nacimiento y fecha de defunción si procede). A su vez un mismo mecenas puede serlo de varios pintores.





Pasamos a tablas las entidades



MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica)

PINACOTECA (nombre, ciudad, direccion, superficie)

ESCUELA (nombre, pais, fecha_aparicion)

Cada tabla la pasamos a SQL

```
CREATE TABLE MECENAS(  
    codigo SMALLINT PRIMARY KEY,  
    nombre VARCHAR(15) NOT NULL,  
    pais VARCHAR(15),  
    fecha_nacimiento DATE,  
    fecha_defuncion DATE  
);
```

MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica)

PINACOTECA (nombre, ciudad, direccion, superficie)

ESCUELA (nombre, pais, fecha_aparicion)

```
CREATE TABLE PINTOR (  
    codigo SMALLINT PRIMARY KEY,  
    nombre VARCHAR(15) NOT NULL,  
    pais VARCHAR(15),  
    ciudad VARCHAR(20),  
    fecha_nacimiento DATE,  
    fecha_defuncion DATE  
);
```

MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica)

PINACOTECA (nombre, ciudad, direccion, superficie)

ESCUELA (nombre, pais, fecha_aparicion)

```
CREATE TABLE CUADRO (  
    codigo SMALLINT PRIMARY KEY,  
    nombre VARCHAR(15) NOT NULL,  
    alto SMALLINT,  
    ancho SMALLINT,  
    fecha DATE,  
    tecnica VARCHAR(20)  
);
```

MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica)

PINACOTECA (nombre, ciudad, direccion, superficie)

ESCUELA (nombre, pais, fecha_aparicion)

```
CREATE TABLE PINACOTECA(  
    nombre VARCHAR(15) PRIMARY KEY,  
    ciudad VARCHAR(20),  
    direccion VARCHAR(100),  
    superficie SMALLINT  
);
```

MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica)

PINACOTECA (nombre, ciudad, direccion, superficie)

ESCUELA (nombre, pais, fecha_aparicion)


```
CREATE TABLE ESCUELA (  
    nombre VARCHAR(15) PRIMARY KEY,  
    pais VARCHAR(15),  
    fecha_aparicion DATE  
);
```

MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

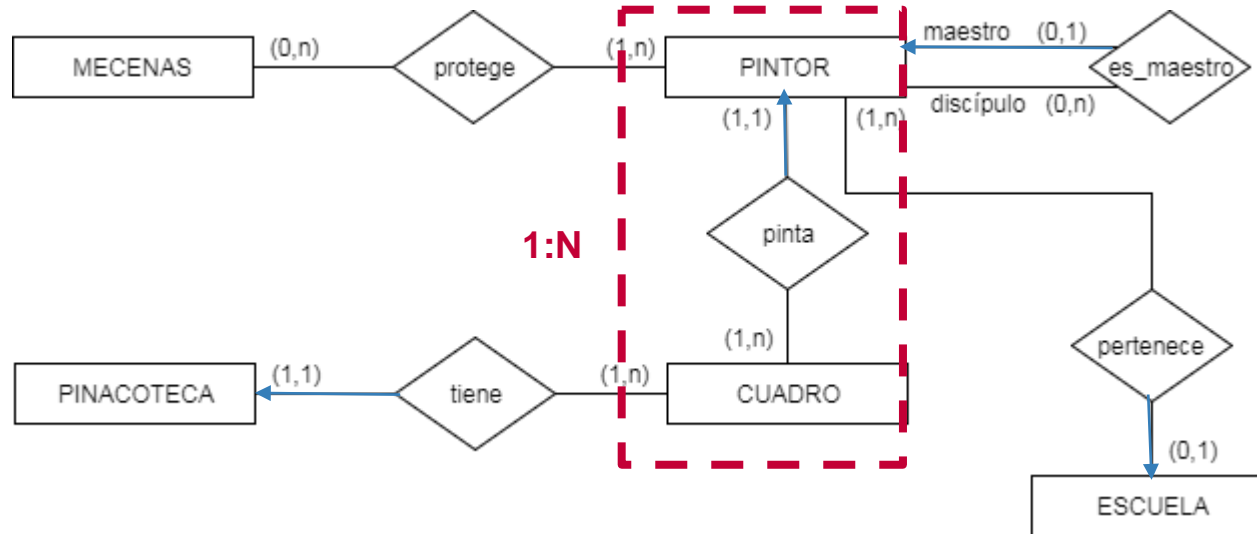
PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica)

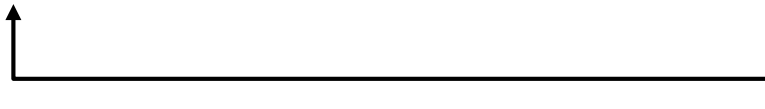
PINACOTECA (nombre, ciudad, direccion, superficie)

ESCUELA (nombre, pais, fecha_aparicion)

Pasamos a tablas las relaciones

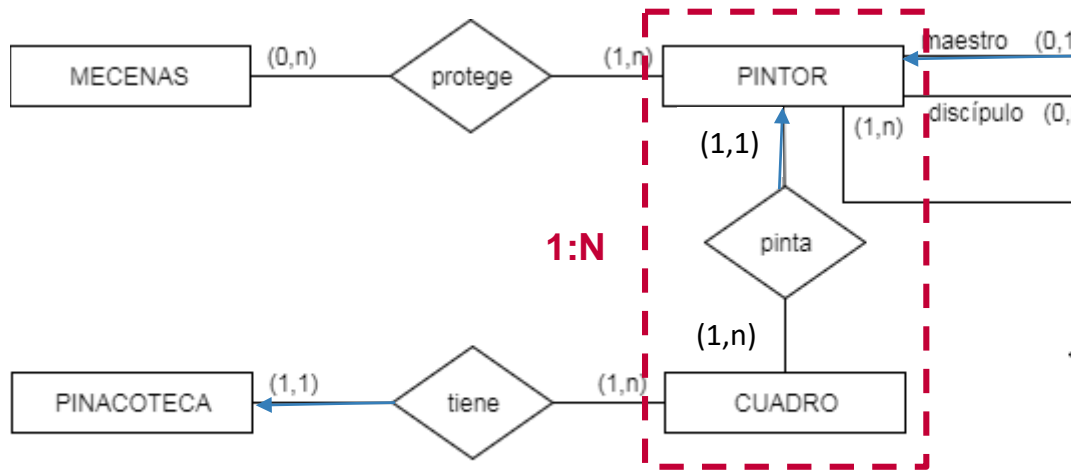


PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)



CUADRO (codigo, nombre, alto, ancho, fecha, tecnica, codigo_pintor)

Pasamos a tablas las relaciones

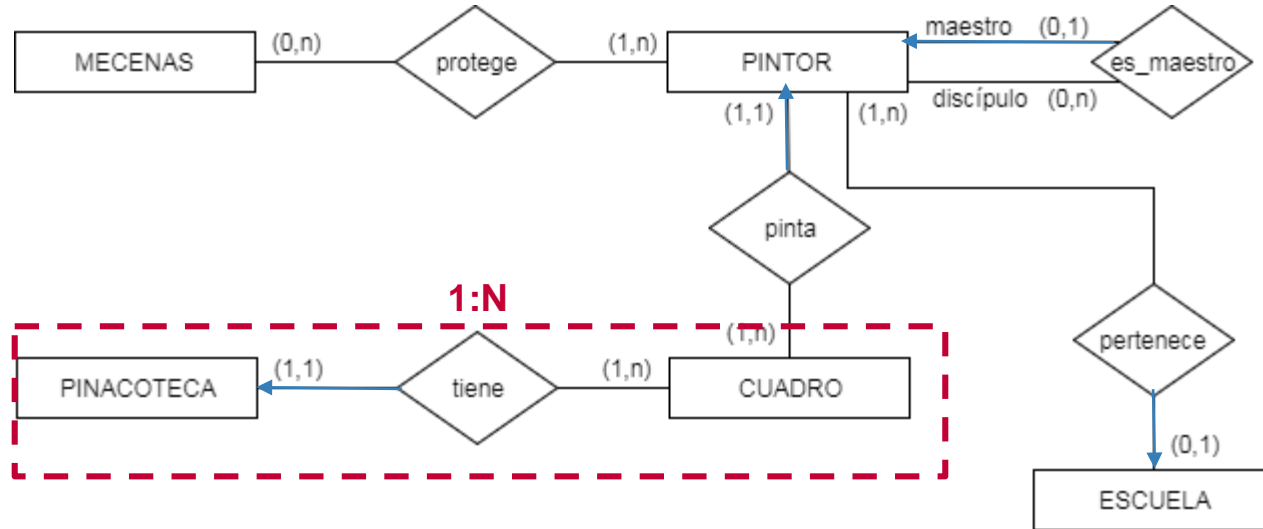


```
CREATE TABLE CUADRO (  
  codigo SMALLINT PRIMARY KEY,  
  nombre VARCHAR(15) NOT NULL,  
  alto SMALLINT,  
  ancho SMALLINT,  
  fecha DATE,  
  tecnica VARCHAR(20),  
  codigo_pintor SMALLINT NOT NULL,  
  FOREIGN KEY (codigo_pintor)  
  REFERENCES PINTOR (codigo)  
  ON DELETE CASCADE A  
  ON UPDATE CASCADE B  
);
```

A Si se borra un pintor se borran sus cuadros.

B Si se cambia la clave primaria de un pintor se actualiza en sus cuadros.

Pasamos a tablas las relaciones



PINACOTECA (nombre, ciudad, direccion, superficie)

CUADRO (codigo, nombre, alto, ancho, fecha, tecnica, nombre_pinacoteca)

```
CREATE TABLE CUADRO (
```

```
  codigo SMALLINT PRIMARY KEY,
```

```
  nombre VARCHAR(15) NOT NULL,
```

```
  alto SMALLINT,
```

```
  ancho SMALLINT,
```

```
  fecha DATE,
```

```
  tecnica VARCHAR(20),
```

```
  codigo_pintor SMALLINT NOT NULL,
```

```
  nombre_pinacoteca VARCHAR(15) NOT NULL,
```

```
  FOREIGN KEY (codigo_pintor) REFERENCES PINTOR (codigo)
```

```
  ON DELETE CASCADE
```

```
  ON UPDATE CASCADE,
```

```
  FOREIGN KEY (nombre_pinacoteca) REFERENCES PINACOTECA (nombre)
```

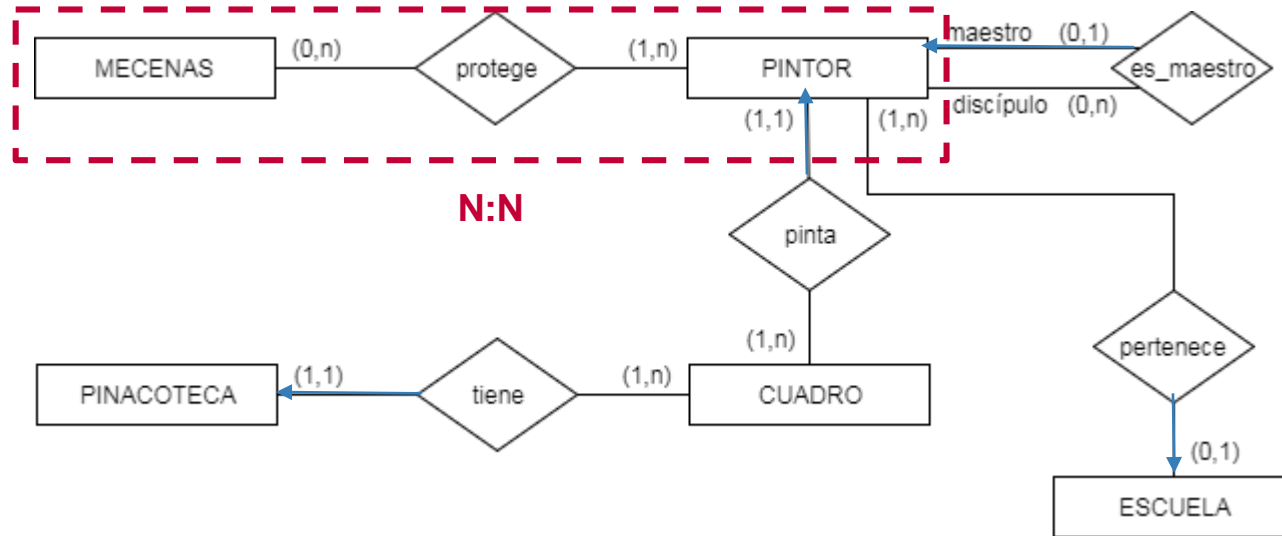
```
  ON DELETE RESTRICT
```

```
  ON UPDATE CASCADE
```

```
);
```

Transformación similar
a la relación “pinta”

Pasamos a tablas las relaciones



PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

PROTEGE (codigo_pintor, codigo_mecenas)

MECENAS (codigo, nombre, pais, fecha_nacimiento, fecha_defuncion)

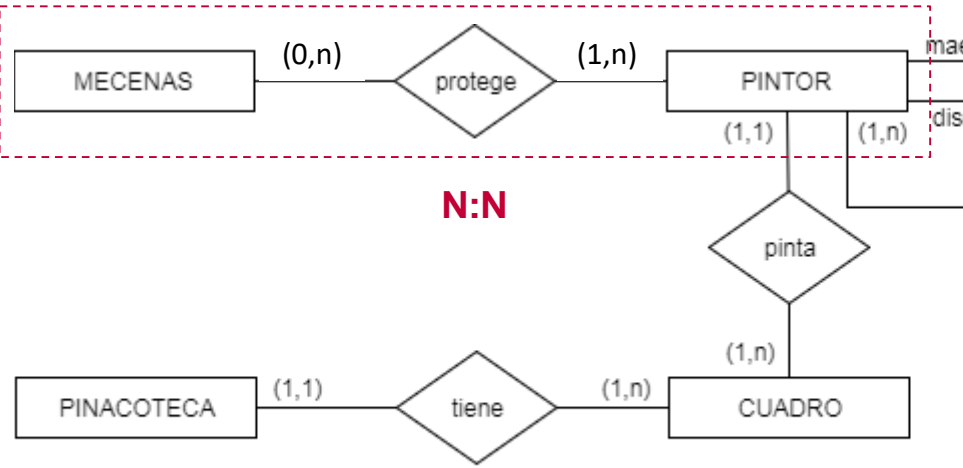
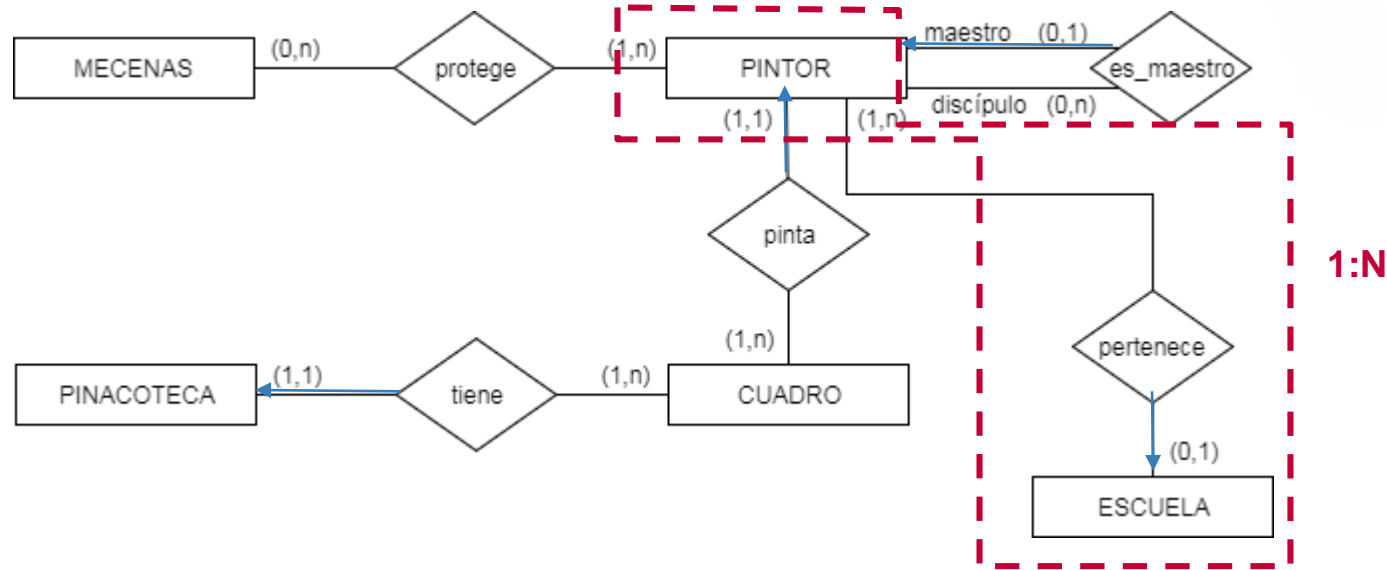


TABLA NUEVA

```
CREATE TABLE PROTEGE(
  codigo_pintor SMALLINT,
  codigo_mecenas SMALLINT,
  PRIMARY KEY (codigo_pintor,
  codigo_mecenas),
  FOREIGN KEY (codigo_pintor)
  REFERENCES PINTOR (codigo)
  ON DELETE CASCADE,
  FOREIGN KEY (codigo_mecenas)
  REFERENCES MECENAS (codigo)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

- A** Si se borra un pintor/mecenas se borra su relación de protección.
- B** Si se cambia la clave primaria de un pintor/mecenas se actualiza en su relación.

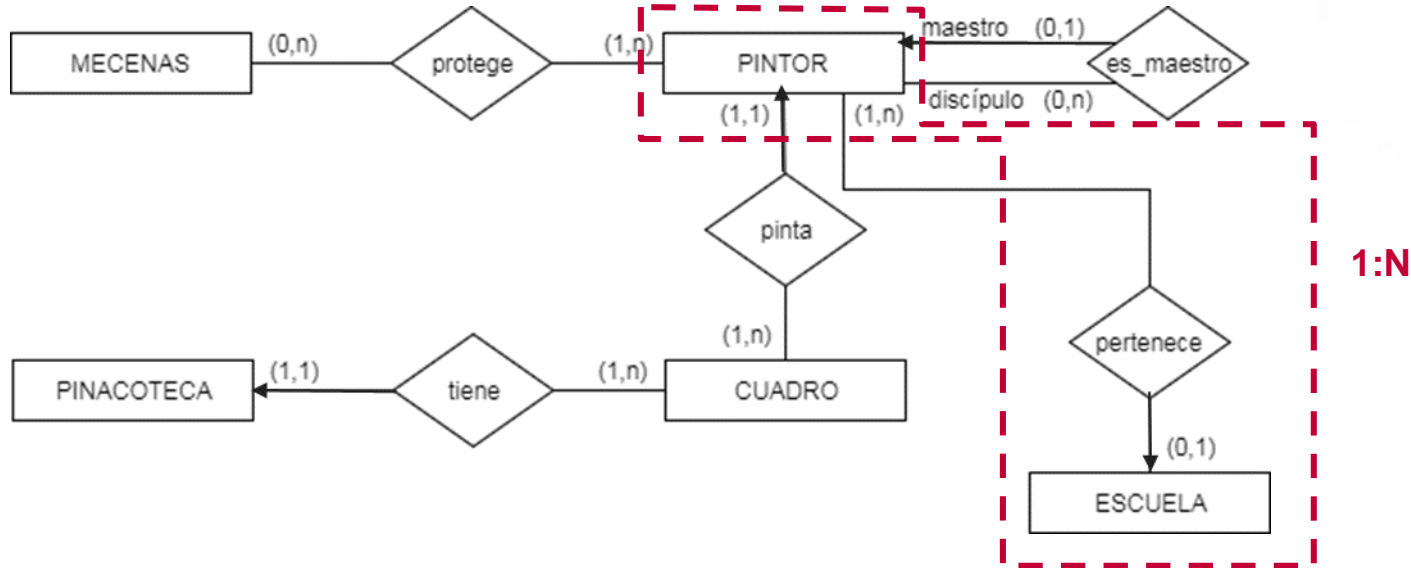
Pasamos a tablas las relaciones



La transformación de la relación PERTENECE se puede hacer de dos maneras:

- Propagando la clave primaria de la tabla ESCUELA a la tabla PINTOR. Esta opción es adecuada si se sabe que la mayoría de los pintores pertenecen a una escuela. Si no es así, esta opción generaría muchos campos a NULL.
- Creando una nueva tabla. Esta opción es adecuada si se sabe que pocos pintores pertenecen a una escuela (se ha elegido esta opción).

Pasamos a tablas las relaciones



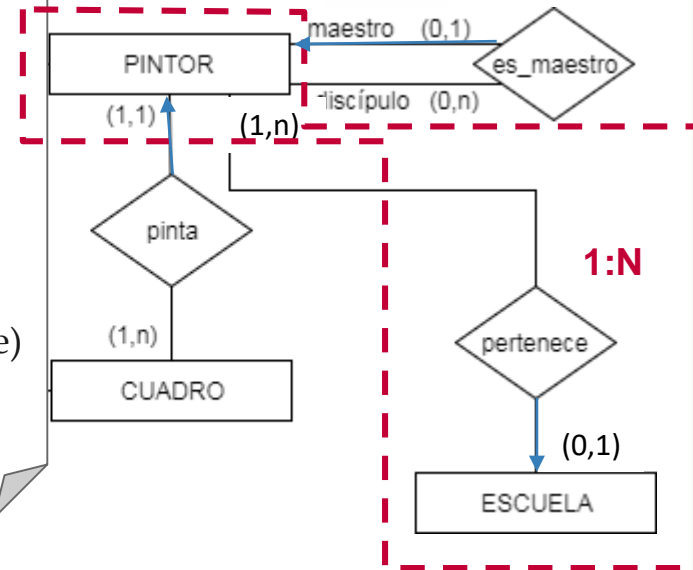
PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion)

PERTENECE (codigo_pintor, nombre_escuela)

ESCUELA (nombre, pais, fecha_aparicion)

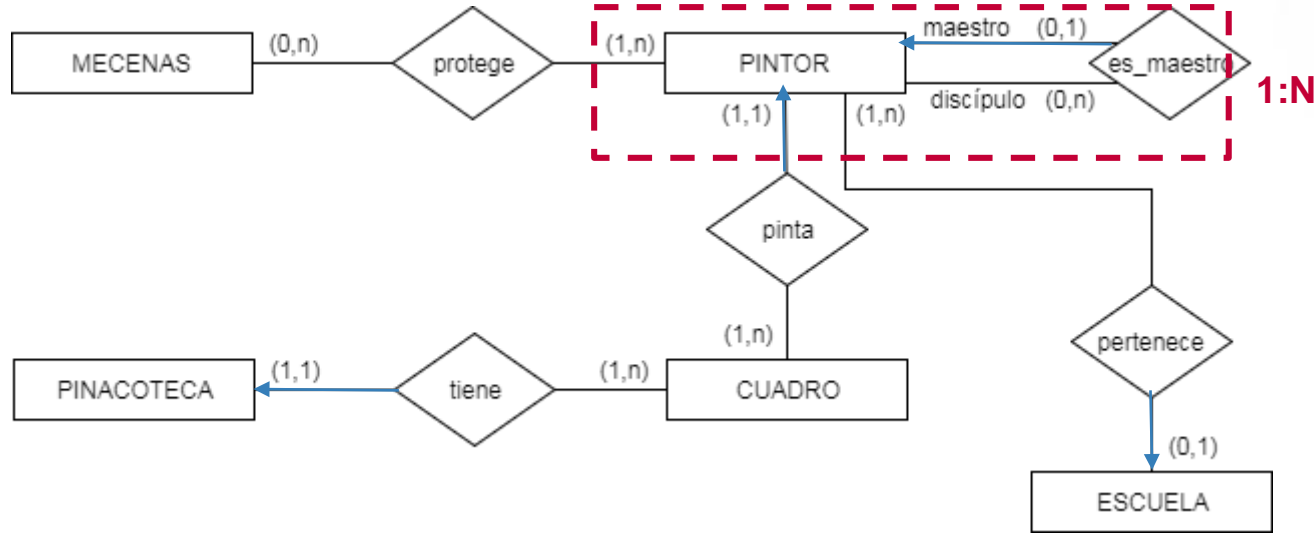
```
CREATE TABLE PERTENECE (
  codigo_pintor SMALLINT PRIMARY KEY,
  nombre_escuela VARCHAR(15),
  FOREIGN KEY (codigo_pintor) REFERENCES PINTOR (codigo)
  ON DELETE CASCADE A
  ON UPDATE CASCADE, B
  FOREIGN KEY (nombre_escuela) REFERENCES ESCUELA (nombre)
  ON DELETE CASCADE A
  ON UPDATE CASCADE B
);
```

TABLA NUEVA



- A** Si se borra un pintor/escuela se borra su relación de pertenencia.
- B** Si se cambia la clave primaria de un pintor/escuela se actualiza en su relación.

Pasamos a tablas las relaciones



La opción de propagar la clave primaria puede hacer que haya valores a NULL. Cuantos menos pintores tengan maestro más valores NULL aparecerán, no obstante, es la opción elegida.

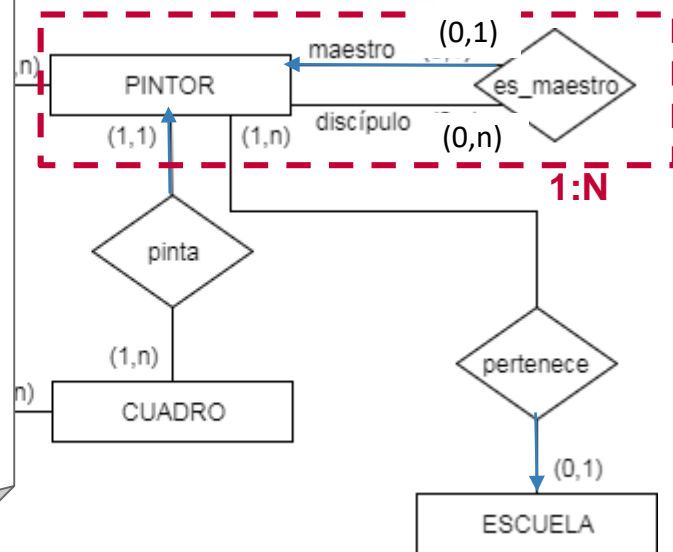
PINTOR (codigo, nombre, pais, ciudad, fecha_nacimiento, fecha_defuncion, codigo_maestro)



```

CREATE TABLE PINTOR (
  codigo SMALLINT PRIMARY KEY,
  nombre VARCHAR(15) NOT NULL,
  pais VARCHAR(15),
  ciudad VARCHAR(20),
  fecha_nacimiento DATE,
  fecha_defuncion DATE,
  codigo_maestro SMALLINT,
  FOREIGN KEY (codigo_maestro) REFERENCES PINTOR (codigo)
  ON DELETE SET NULL A
  ON UPDATE CASCADE B );

```



- A** Si se borra un pintor-maestro se pone a NULL su correspondiente clave ajena.
- B** Si se cambia la clave primaria de un pintor se actualiza su correspondiente clave ajena.

