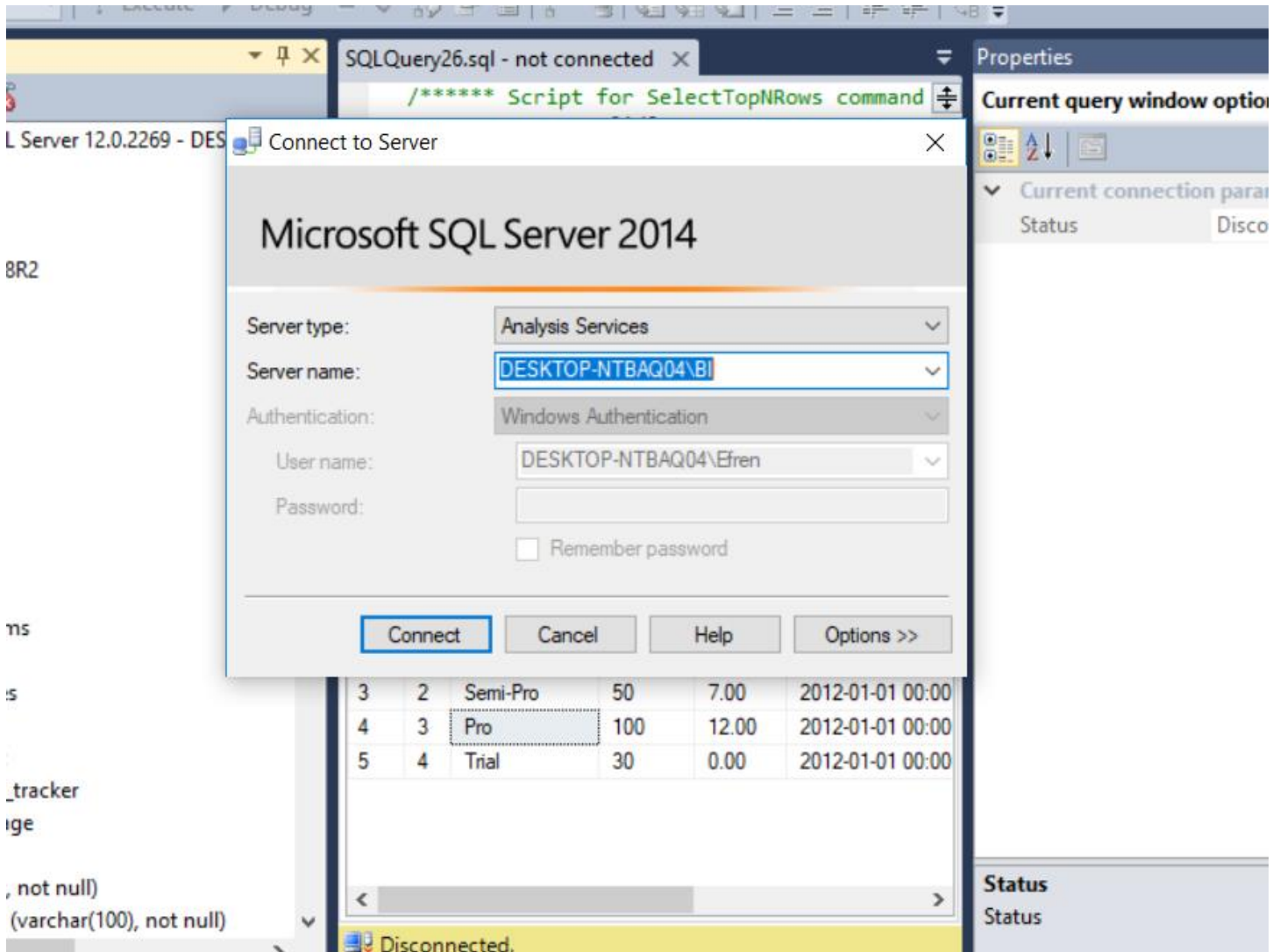


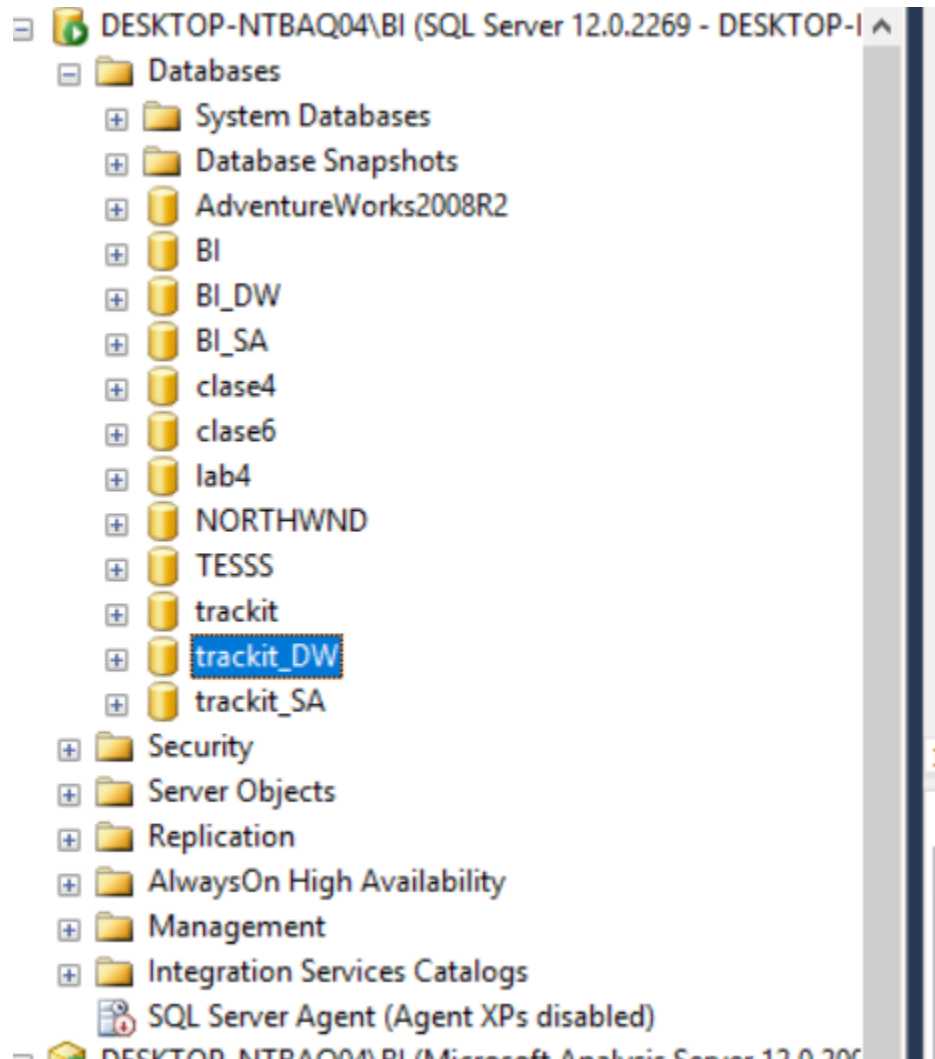
Crear, usar y modificar una BD en SQL SERVER

A continuación, se van a explicar los pasos a realizar para construir y utilizar una base de datos sencilla.

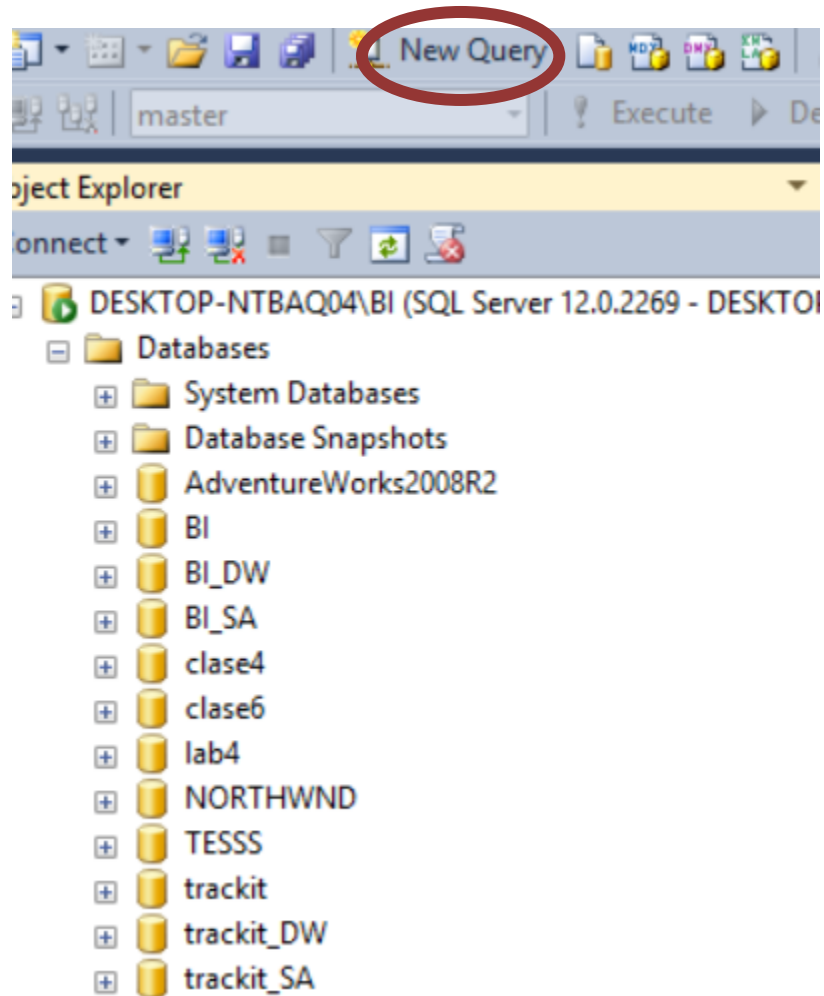
- **Paso 1.** Abrir el SQL Server Management Studio y conectarse como administrador



- **Paso 2.** Una vez conectado en la parte izquierda se muestran las BD alojadas en el servidor, por defecto las del sistema () y otras si se han creado previamente.



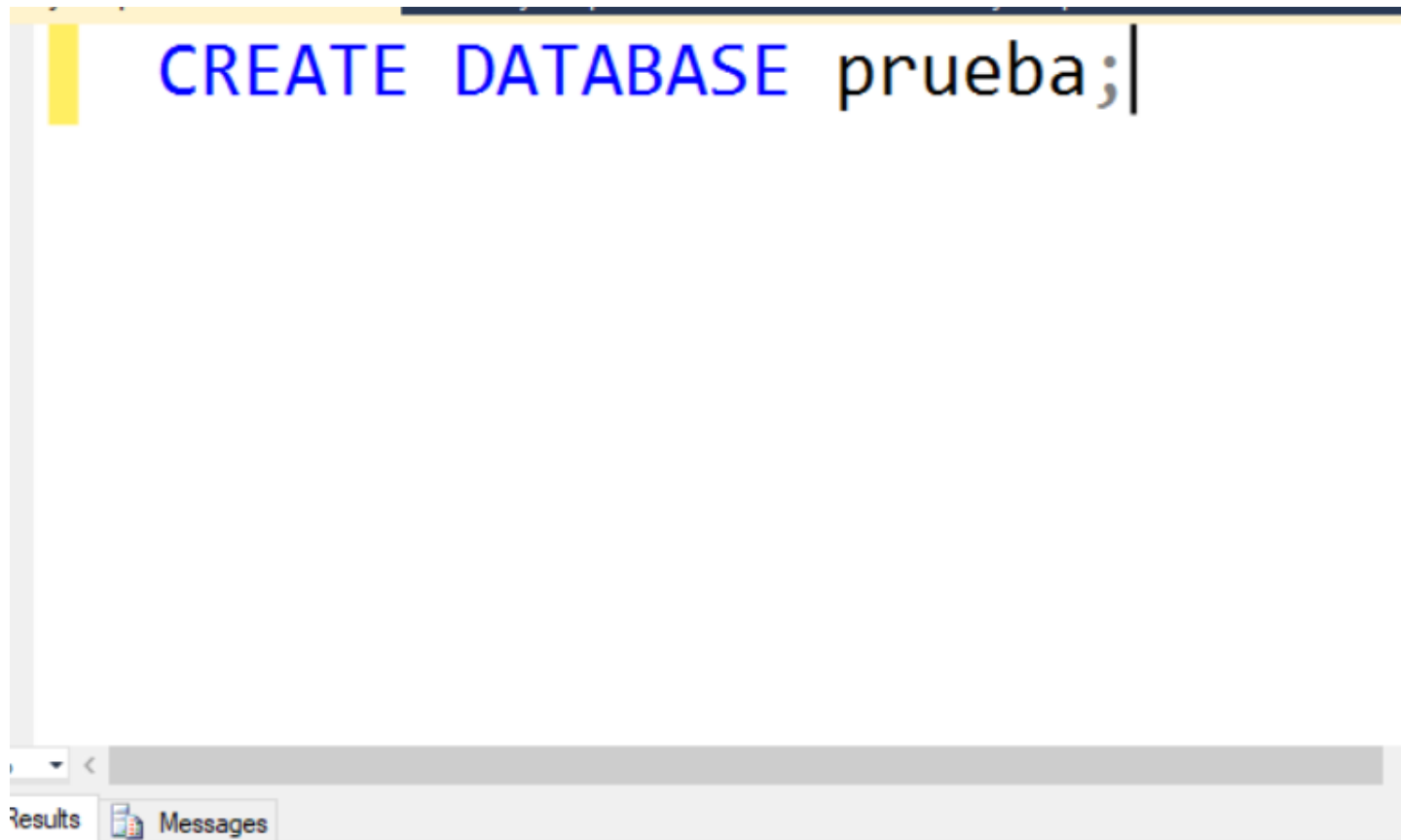
- **Paso 3.** Abrir una consulta en la conexión actual



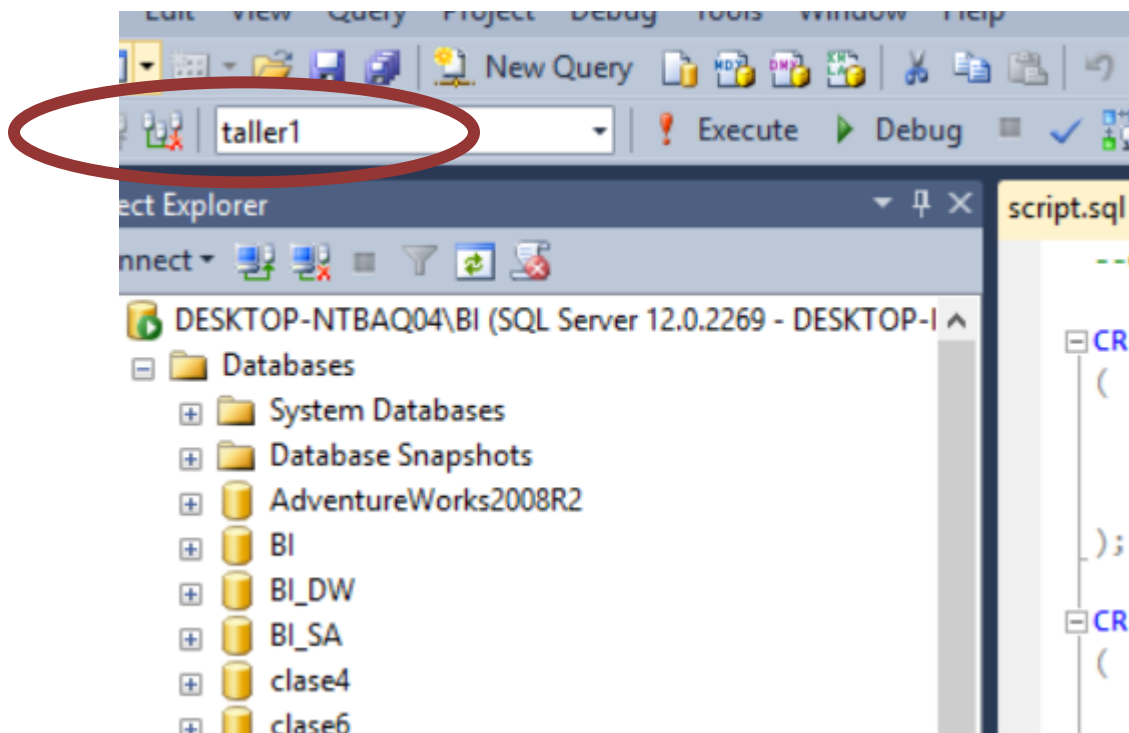
- **Paso 4.** Ejecutar las siguientes instrucciones. Recordar que en el estándar la instrucción termina con punto y coma
--Instrucción para **crear** la base de datos

CREATE DATABASE prueba;

--Instrucción para **seleccionar** la base de datos a utilizar

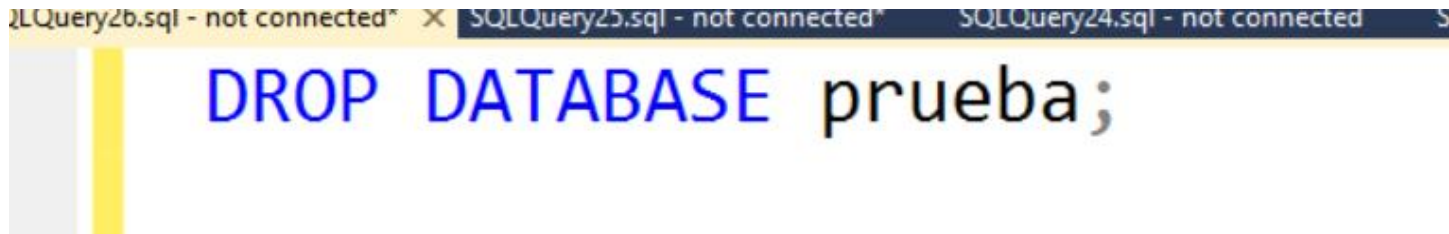


Seleccionar la base de datos a trabajar.



--Instrucción para **eliminar** una base de datos

DROP DATABASE prueba;



Crear, modificar y borrar una tabla en SQL SERVER

A continuación, se van a explicar los pasos a realizar para construir, modificar y eliminar tablas en una base de datos.

- **Paso 1.** Crear una base de datos, o utilizar una creada previamente
- **Paso 2.** Abrir una consulta en la conexión actual
- **Paso 3.** Para crear una tabla ejecutar las siguientes instrucciones

CREATE TABLE persona

(

--si la llave es compuesta se indica así **primary key (atributo1, atributo 2)**

cedula CHAR(11) NOT NULL PRIMARY KEY,

nombre VARCHAR(50) NOT NULL,

apellido1 VARCHAR(50) NOT NULL,

apellido2 VARCHAR(50) NOT NULL,

correo VARCHAR(100) NOT NULL,

telefono CHAR(9) NULL,

direccion VARCHAR(200) NOT NULL

);

- **Paso 4.** Para modificar una tabla se ejecutan las siguientes instrucciones

--Agregar una columna como llave primaria

ALTER TABLE persona ADD CONSTRAINT pk_cedula_persona PRIMARY KEY (cedula) ;

--Agregar una nueva columna a la tabla y que no pueda ser nula

ALTER TABLE persona ADD fecha_nacimiento DATETIME NOT NULL;

--Borrar una columna de una tabla

ALTER TABLE persona DROP COLUMN fecha_nacimiento;

--Agregar una nueva columna a la tabla

ALTER TABLE persona ADD prueba CHAR(10) NOT NULL;

--Modificar el tamaño de la columna que acabamos de agregar

ALTER TABLE persona ALTER COLUMN prueba CHAR(12) NOT NULL;

--Borrar la columna

ALTER TABLE persona DROP COLUMN prueba;

--Borrar la tabla

DROP TABLE persona;



- **Paso 5.** Cree las siguientes tablas en la base de datos

```
CREATE TABLE estudiante
(
    cedula CHAR (11) NOT NULL,
    carnet INT NOT NULL,
    nombre VARCHAR (50)
);
```

```
ALTER TABLE estudiante
ADD CONSTRAINT pk_cedula_estudiante PRIMARY KEY (cedula);
```

```
ALTER TABLE estudiante
ADD CONSTRAINT fk_cedula_ FOREIGN KEY (cedula) REFERENCES persona;
```

```
ALTER TABLE estudiante
ADD CONSTRAINT unique_carnet UNIQUE (carnet);
```

```
CREATE TABLE profesores
(
    cedula CHAR (11) NOT NULL,
    especialidad VARCHAR (50) NOT NULL
);
```

```
ALTER TABLE profesores
ADD CONSTRAINT pk_cedula PRIMARY KEY (cedula);
```

```
CREATE TABLE carrera
(
    --serial(1,1) inicia el identificador en 1 y aumenta automáticamente el valor en 1
    id_carrera int NOT NULL identity,
    nombre VARCHAR(50) NOT NULL,
    CONSTRAINT pk_idcarrera PRIMARY KEY (id_carrera)
);
```

--Preguntar al professor por que esta linea da error

```
ALTER TABLE carrera
ADD CONSTRAINT pk_carrera PRIMARY KEY (id_carrera);
```

- **Paso 6.** Complete la creación de las siguientes tablas. Agregue las llaves primarias y foráneas que sean necesarias

```
create table programa (
    id_programa
```



```
        fecha
        estado
    );
```

```
create table curso (
```

```
    codigo
    cedula
    id_programa
    nombre
    creditos
    tipo
    periodo
    fecha
    estado
);
```

```
create table contenido (
```

```
    id_contenido
    id_curso
    contenido
);
```

```
create table grupo (
```

```
    id_grupo
    id_curso
    numero_grupo
);
```

```
create table grupo_estudiante (
```

```
    id_grupo
    carne
    nota
    estado
);
```

- **Paso 7.**Realice los siguientes ejercicios.

- Agregue a la tabla **programa** un atributo descripción
- Modifique el atributo especialidad de la tabla **profesor** para que sea un varchar de 200 caracteres.
- Agregue un atributo **id_carrera** a la tabla **programa**. Luego, convierta este atributo en una llave foránea a la tabla **carrera**.
- Agregue un atributo cupo a la tabla **grupo**.

Sede San Carlos

Prof. Efren Jiménez Delgado

Curso: ISW-312

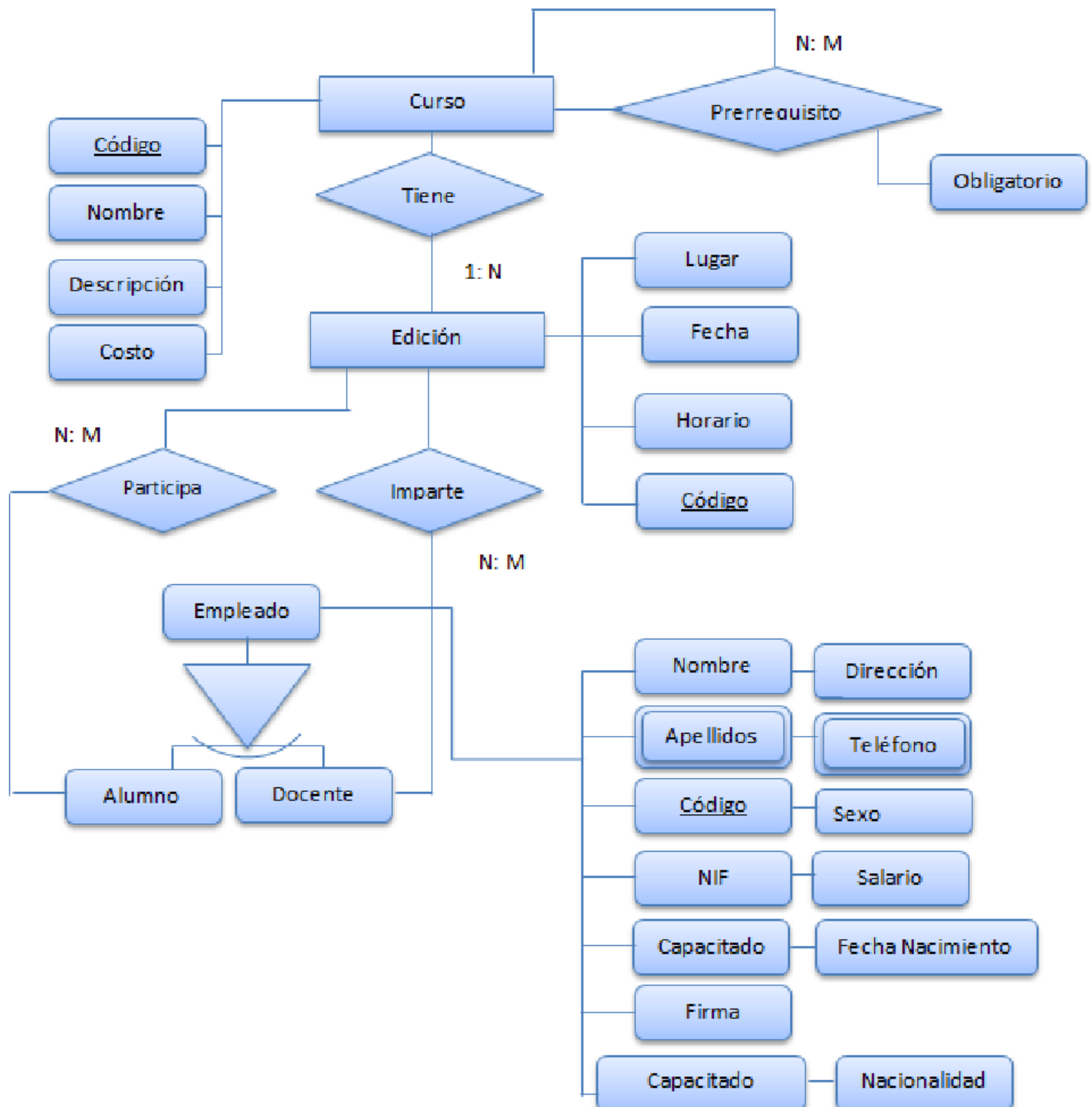


Fundamentos de base de datos

- o Elimine el atributo **estado** de la tabla **grupo_estudiante**.
- o Cambie el atributo nombre de la tabla **estudiante** para que sea **NOT NULL**
- o Borre la tabla **contenido**

Modelo Relacional – Práctica

Recuerde que en el transcurso de las clases hemos trabajado con el siguiente modelo E-R.





A partir del modelo anterior generamos el siguiente modelo relacional. Con dicho modelo relacional, elabore la base de datos en **Postgresql**. Recuerde que debe definir las llaves primarias y foráneas necesarias. Así como, definir el tipo de datos adecuado para cada atributo.

curso (código, nombre, descripción, costo)

prerrequisito (id, código1, código2, obligatorio)

edición (código, lugar, fecha, horario, código_curso)

empleado (código, nombre, dirección, apellido1, apellido2, sexo, cedula, salario, capacitado, nacionalidad, fecha_nacimiento, es_alumno, es_profesor)

telefono (id, código, telefono)

participa (código_edición, código_empleado)

imparte (código_edición, código_empleado)

Al finalizar, para revisar el ejercicio anterior cree un nuevo diagrama de la base de datos, dando clic sobre la base de datos que está trabajando y luego agregue las tablas deseadas, debe obtener un diagrama similar al anterior.

