

Actividad de aprendizaje 3-02

Se trata de:

- Continuar añadiendo tablas a la base de datos **centro** creada en la actividad de aprendizaje 3-02.

Objetivos:

- Crear tablas.
- Eliminar o borrar tablas.
- Establecer tipos de datos en las columnas de las tablas.
- Establecer modificadores en los tipos de datos.
- Establecer claves primarias.
- Establecer índices.
- Establecer claves ajenas.

Procedimiento

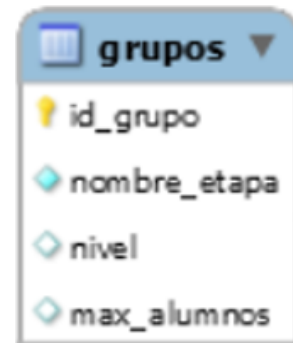
Esta actividad debes realizarla con el cliente **mysql** conectando al servidor MySQL en local.

1.- Activa o abre la base de datos **CENTRO**.

USE CENTRO;

2.- Crea una tabla **GRUPOS** que contiene todos los grupos que hay en el centro. Cada grupo tiene un identificador de 4 caracteres, el nombre del ciclo formativo o etapa al que pertenece, el nivel y el número máximo de alumnos que tiene autorizado.

```
CREATE TABLE grupos (  
id_grupo CHAR(4) NOT NULL,  
nombre_etapa VARCHAR(80) NOT NULL,  
nivel TINYINT,  
max_alumnos TINYINT,  
PRIMARY KEY(id_grupo));
```



3.- En la actividad anterior se creó la tabla alumnos que tiene una clave ajena relacionada con la tabla **GRUPOS**. Se podría usar la instrucción **ALTER TABLE** para añadir la relación de clave ajena en la tabla alumnos. Pero no lo vamos a hacer así, vamos a crear nuevamente la tabla **ALUMNOS** declarando todas sus columnas y añadiendo la restricción de clave ajena.

3.a.- Elimina la tabla **ALUMNOS**.

DROP TABLE ALUMNOS;

3.b.- Crea la tabla **ALUMNOS** en las mismas condiciones que en la actividad 3-02 y añadiendo la restricción de clave ajena.

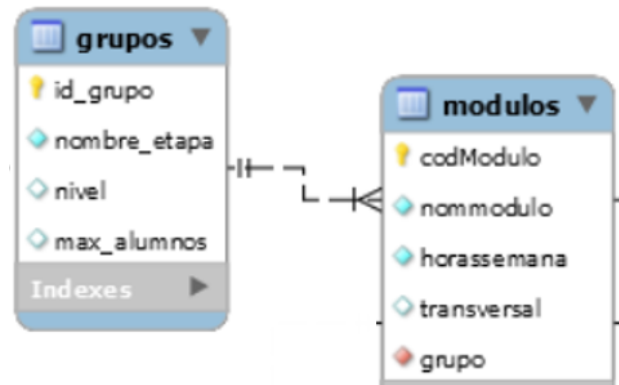
```
CREATE TABLE alumnos (  
  numalumn INT(3) UNSIGNED ZEROFILL NOT NULL  
  AUTO_INCREMENT,  
  nombre VARCHAR(15) NOT NULL,  
  apellidos VARCHAR(25) NOT NULL,  
  direccion VARCHAR(40),  
  dni CHAR(9) NOT NULL,  
  grupo CHAR(4),  
  PRIMARY KEY(numalumn),  
  UNIQUE(dni),  
  CONSTRAINT fk_grupo_alumnos FOREIGN KEY (grupo) REFERENCES  
  grupos(id_grupo));
```



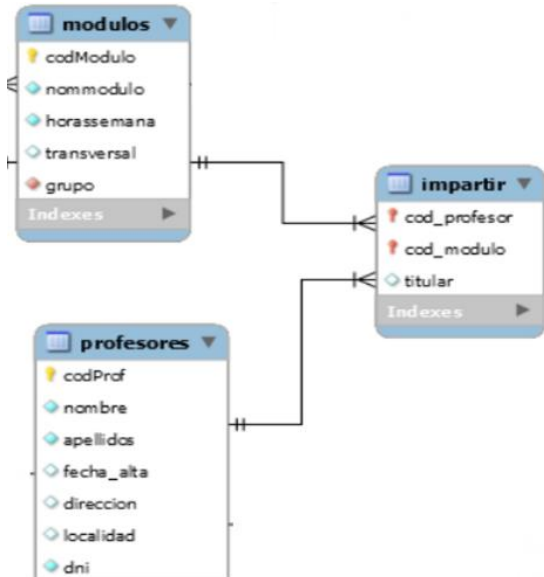
4.- Para la tabla **MODULOS** haz lo mismo que con **ALUMNOS**, elimínala primero y vuélvela a crear añadiendo la restricción de clave ajena.

DROP TABLE modulos;

```
CREATE TABLE modulos (  
  codmodulo CHAR(6) not null,  
  nommodulo VARCHAR(80),  
  horassemana TINYINT NOT NULL,  
  transversal BOOLEAN DEFAULT false,  
  grupo CHAR(4) NOT NULL,  
  PRIMARY KEY(codmodulo),  
  CONSTRAINT fk_grupo_modulo FOREIGN KEY (grupo) REFERENCES  
  grupos(id_grupo));
```

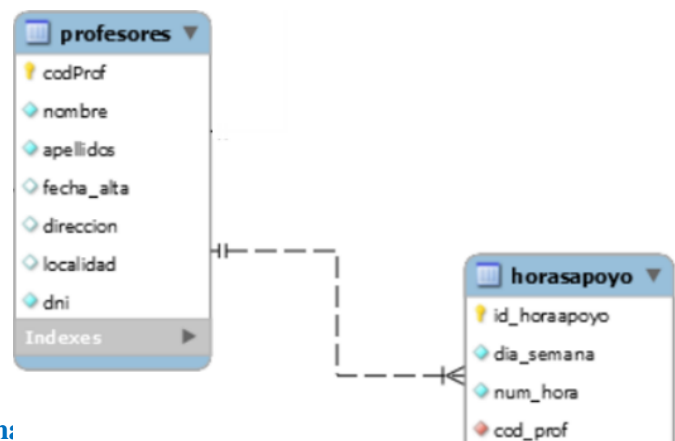


5.- Crea la tabla **IMPARTIR** que contiene las columnas **código de profesor**, **código de módulo**, **titular** (para indicar si el profesor actúa como titular o como profesor de apoyo). Establece que código de profesor y código de módulo son clave principal, código de profesor es clave ajena con borrado y modificación en cascada, al igual que código de módulo, por defecto se establece que un profesor sería titular.



```
CREATE TABLE impartir (
    cod_profesor INT UNSIGNED,
    cod_modulo CHAR(6) not null,
    titular BOOLEAN DEFAULT true,
    PRIMARY KEY(cod_profesor,cod_modulo),
    CONSTRAINT fk_profesor_impartir FOREIGN KEY (cod_profesor) REFERENCES
    profesores(codprof) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT fk_modulo_impartir FOREIGN KEY (cod_modulo) REFERENCES
    modulos(codmodulo) ON UPDATE CASCADE ON DELETE CASCADE);
```

6.- Crea la tabla **HORASAPOYO** que va a contener las horas en que trabajan los profesores ayudantes. Esta tabla tendrá las columnas código de profesor, día de la semana (TINYINT al igual que en horario), hora (TINYINT al igual que en horario). Código de profesor es clave ajena con borrado y modificación en cascada. Los elementos de esta tabla tendrán un identificador numérico y autoincrementado. Establece que el conjunto de columnas día, hora y código de profesor no admitan valores repetidos.

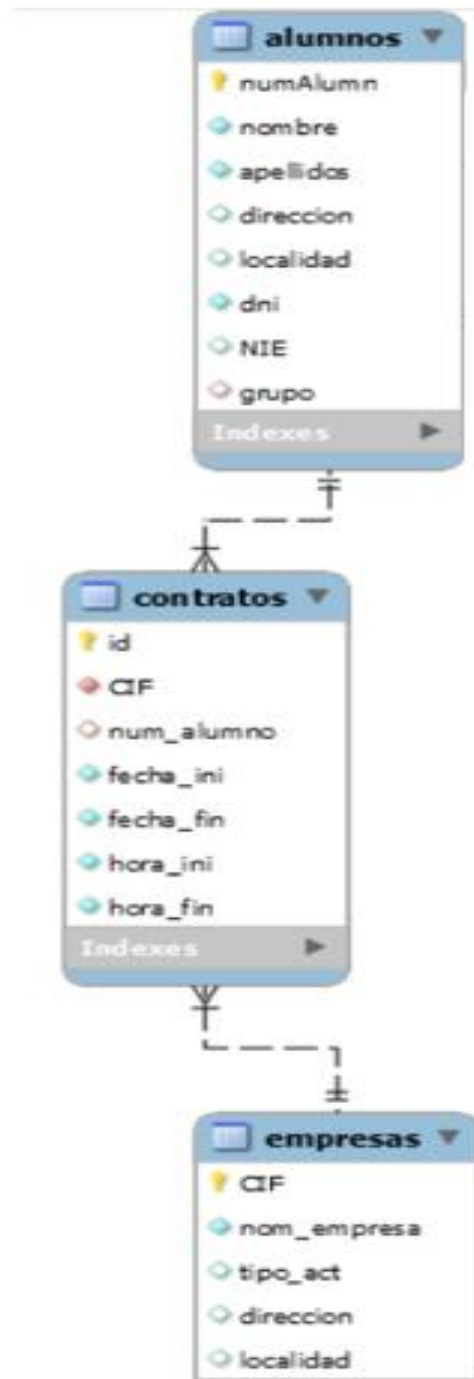


```
CREATE TABLE horasapoyo (
    id_horaapoyo INT NOT NULL
    AUTO_INCREMENT,
    cod_prof INT UNSIGNED NOT NULL,
    dia_semana TINYINT NOT NULL,
    num_hora TINYINT NOT NULL,
    PRIMARY KEY(id_horaapoyo),
    UNIQUE uk_hora_horasapoyo (dia_semana,
    num_hora),
    CONSTRAINT fk_profesor_horasapoyo FOREIGN KEY (cod_prof) REFERENCES
    profesores(codprof) ON UPDATE CASCADE ON DELETE CASCADE);
```

7.- Crea la tabla **CONTRATOS** que contiene los datos sobre la realización del módulo FCT por los alumnos en las empresas. Cada contrato contiene las columnas **CIF**

de la empresa, dni de alumno, número de alumno, fecha de comienzo de las prácticas de FCT, fecha finalización, hora inicial trabajo, hora final trabajo, apto. CIF es clave ajena al igual que número de alumno. Se usa como identificador de cada contrato un número entero autoincrementado. No se debe permitir el borrado o la modificación en EMPRESAS o en ALUMNOS si hay filas relacionadas en CONTRATOS.

```
CREATE TABLE contratos (  
  id INT NOT NULL AUTO_INCREMENT,  
  cif CHAR(9) NOT NULL,  
  dni CHAR(9) NOT NULL,  
  num_alumno INT UNSIGNED,  
  fecha_ini DATE NOT NULL,  
  fecha_fin DATE,  
  hora_ini TIME NOT NULL,  
  hora_fin TIME NOT NULL,  
  apto BOOLEAN,  
  PRIMARY KEY(id),  
  CONSTRAINT fk_numalumno_contratos  
  FOREIGN KEY (num_alumno)  
  REFERENCES alumnos(numalumn),  
  CONSTRAINT fk_cifempresa_contratos  
  FOREIGN KEY (cif)  
  REFERENCES empresas(cif));
```



8.- Crea la tabla **TITULACIONES**.

```
CREATE TABLE titulaciones (  
  cod_profesor INT UNSIGNED NOT NULL  
  AUTO_INCREMENT,  
  num_tit TINYINT NOT NULL,  
  tipo_tit ENUM('Licenciatura','Diplomatura','Ingeniería Técnica','Ingeniería  
    Superior') NOT NULL DEFAULT 'Licenciatura',  
  nombre_tit VARCHAR(30) NOT NULL,  
  PRIMARY KEY(cod_profesor,num_tit),  
  CONSTRAINT fk_profesor_titulaciones FOREIGN KEY (cod_profesor)  
    REFERENCES profesores(codprof));
```

