## Práctica realizada por:

U3P04: Creación de bases de datos

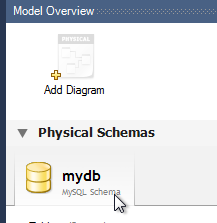
Una vez familiarizados con MySQL Server y MySQL Workbench, vamos a crear bases de datos y a introducir datos en ellas. Consulta el documento “Tipos de datos en MySQL” si lo necesitas.

Entrega este documento sin borrar el enunciado, introduciendo capturas donde se indique con este color. Entrega en PDF.

# 1. Creación de un modelo de bases de datos sencillo con coches

1.1. Arranca MySQLWorkbench y crea un nuevo Modelo EERR llamado Coches\_Nombre (donde nombre es tu nombre).

Asegúrate de poner el nombre también en “Schema”, para que la base de datos que generaremos se llame así y no “mydb”. Para ello haz clic con el botón secundario en “mydb” y edita su nombre:



1.2. Haz clic en “Add Diagram” y comienza tu diseño añadiendo una tabla “Coches”. En el diseño incluye una única tabla Coches con estos campos:

* Modelo, de tipo texto con un máximo de 45 caracteres
* Potencia, un número entero, sabiendo que los valores serán siempre menores de 100.000.
* Fecha, de tipo fecha sin hora
* Precio, un número que puede llevar decimales
* Disponible: de tipo booleano (true o false, lee el documento de tipos de datos)
* Observaciones, un texto de un máximo de 80 caracteres

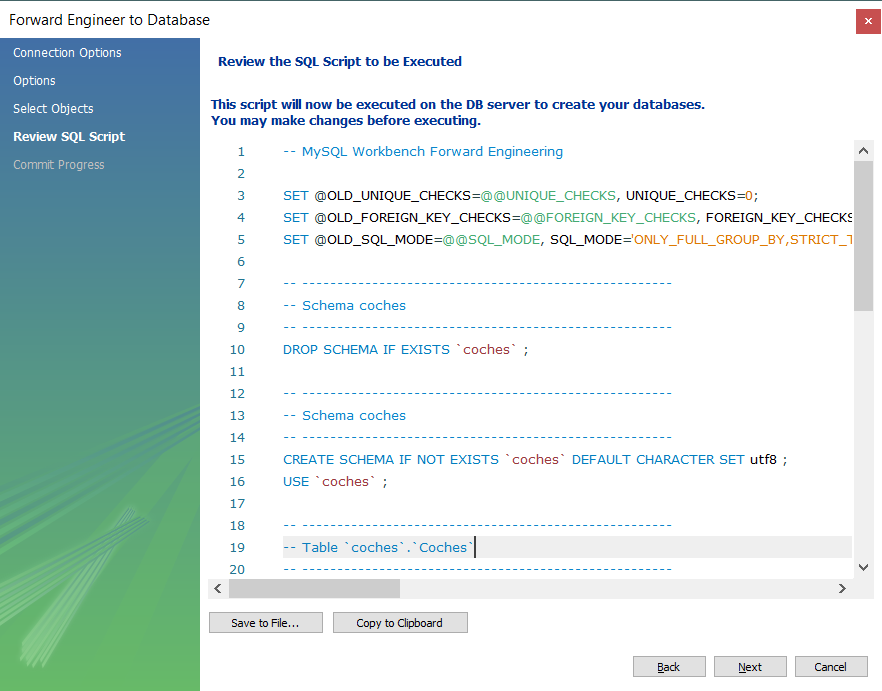
Captura del diseño (no borres los mensajes de este color):

1.3. En el menú Database 🡪 Forward Engineer genera un script de creación de tu base de datos CochesNombre y ejecútalo en tu servidor. Marca las siguientes opciones para generar el script:

* DROP objects before each CREATE object: esto sirve para que puedas ejecutar
* Generate DROP SCHEMA

Estas opciones harán que el script incluya unas órdenes previas para eliminar la base de datos actual antes de crearla de nuevo. Es útil para poder ejecutar el script varias veces sin que MySQL nos diga que no puede crear los elementos porque ya existen.

Captura de la pantalla en que sale el script ya generado para su revisión. Sustituye la siguiente captura por la tuya:



Observa que puedes guardar en un archivo este script por si algo sale mal: es aconsejable (aunque no obligatorio) hacerlo.

1.4. Haz clic en “Siguiente” para añadir la base de datos a tu servidor. La podrás ver en el panel de navegación de MySQL Workbench, pero es posible que tengas que hacer clic con el botón secundario y escoger “Refresh”.

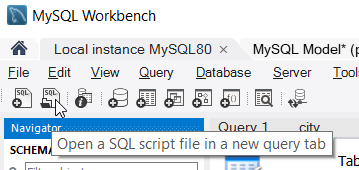
Captura del panel izquierdo de navegación en MySQL Workbench donde aparezca tu nueva base de datos

1.5. Comprueba que se han creado los campos que pusiste en el diagrama.

# 2. Inserción de datos a través de un script SQL

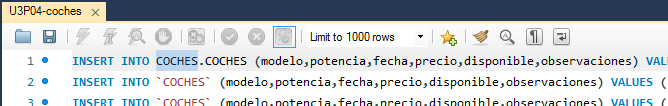
2.1. Abre con Notepad++ el archivo SQL suministrado con la práctica. Este archivo es un script de sentencias SQL que añaden datos a la tabla. Estudiaremos este tipo de sentencias a lo largo del curso, pero de momento nos basta con comprender lo que hacen. Observa que como valores para “disponible” se introduce a veces “1” o “0” y a veces “true” o “false”.

2.2. Haz clic en el icono que permite abrir un script SQL:

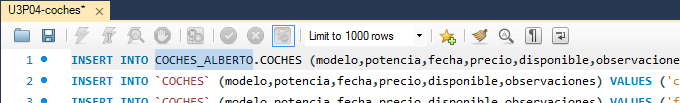


Carga el archivo suministrado con la práctica y observa:

* MySQL no es sensible a mayúsculas. Da igual si hablamos de la base de datos COCHES o coche, y da igual si hablamos del campo MODELO o Modelo.
* Observa que para “disponible” se han almacenado todos los valores como 1 ó 0, a pesar de que en algunas líneas del script se indicaba “true” o “false”. Sin embargo, si intentas escribir manualmente “true” o “false” en la base de datos, obtendrás un error.
* Debes hacer una modificación, ya que el script está pensado para que la base de datos se llame COCHES:



Sustituye el nombre de la base de datos por el tuyo, por ejemplo COCHES\_ALBERTO:



Ten en cuenta que la tabla se sigue llamando COCHES así que la tabla no hay que cambiarla.

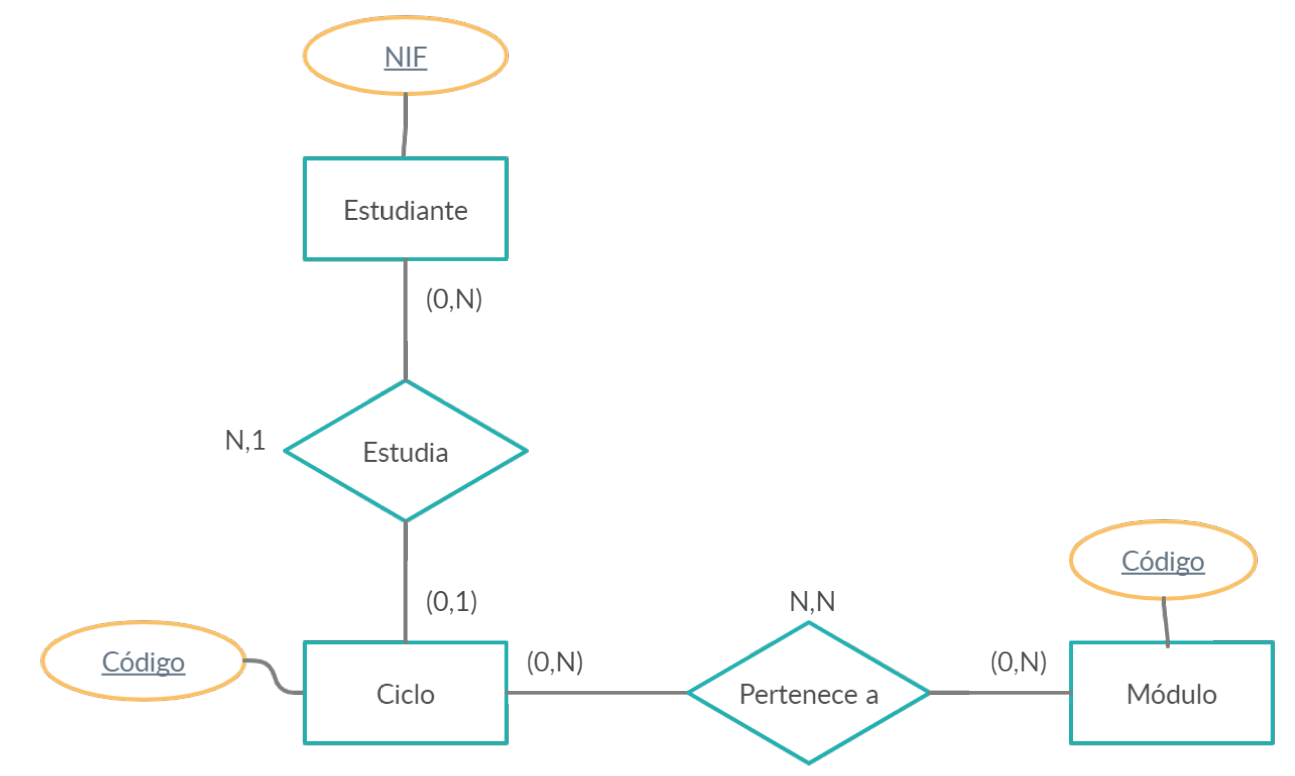
2.3. Ejecuta el script con el icono del rayo y revisa el panel inferior por si hubiese errores.

2.4. Comprueba que se han introducido los datos correctamente.

# 3. Diseño e implementación de una base de datos sencilla con relaciones

## 3.1. Modelado del problema

Vamos a considerar el ejercicio que utilizamos en clase para implementar un diseño de bases de datos en Access:



En el diagrama sólo se indican las claves primarias de las entidades. El resto de atributos los puedes personalizar tú, pero encontrarás una propuesta en el siguiente paso a relciopodemos proponer algu:

* Estudiante: NIF, Nombre, Apellidos, Nacionalidad, Fecha\_nacimiento
* Ciclo: Código (por ejemplo SMR), Nombre, Tipo (medio o superior)
* Módulo: Código (por ejemplo FOL), Nombre, Horas\_semanales

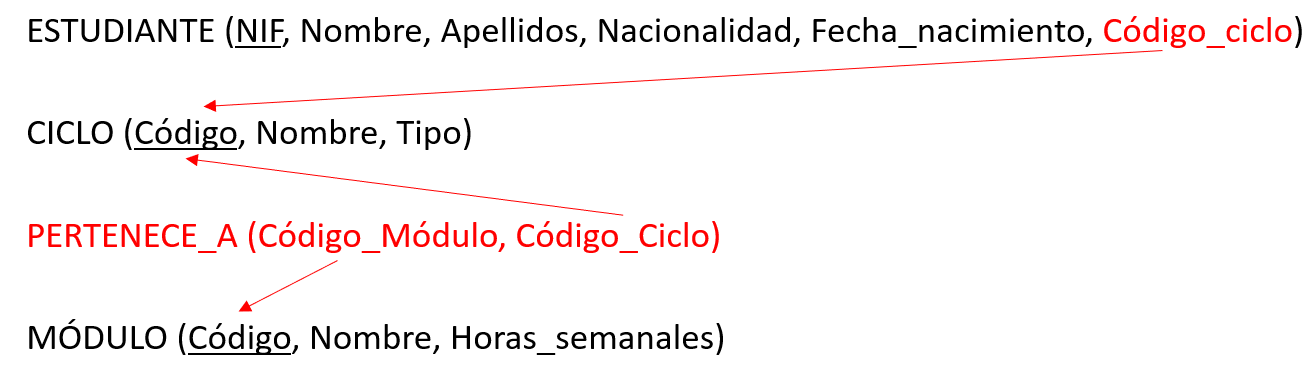
3.1.1. Crea un nuevo modelo desde el menú principal de MySQL Workbench, y llámalo Ciclos.

3.1.2. Crea las tablas correspondientes a las tres entidades, sin hacer aún el paso a relacional. El resultado debería ser algo así:



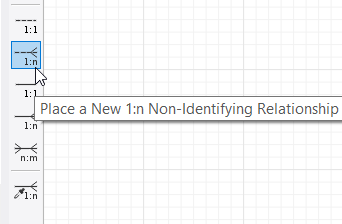
3.1.3. Ahora hacemos el paso a relacional, quedando:

* Relación “Estudia” 🡪 Se propagará el código de ciclo como clave ajena a la tabla estudiante
* Relación “Pertenece a” 🡪 Se creará una nueva tabla “Pertenece”, o “MóduloPerteneceACiclo”

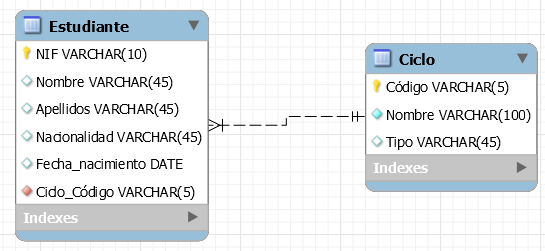


## 3.2. Modelado de la relación N:1 “Estudia”

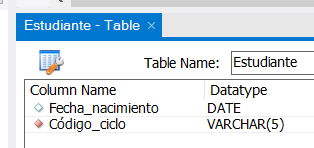
3.2.1.Haz clic en el botón de relación sin dependencia 1:N:



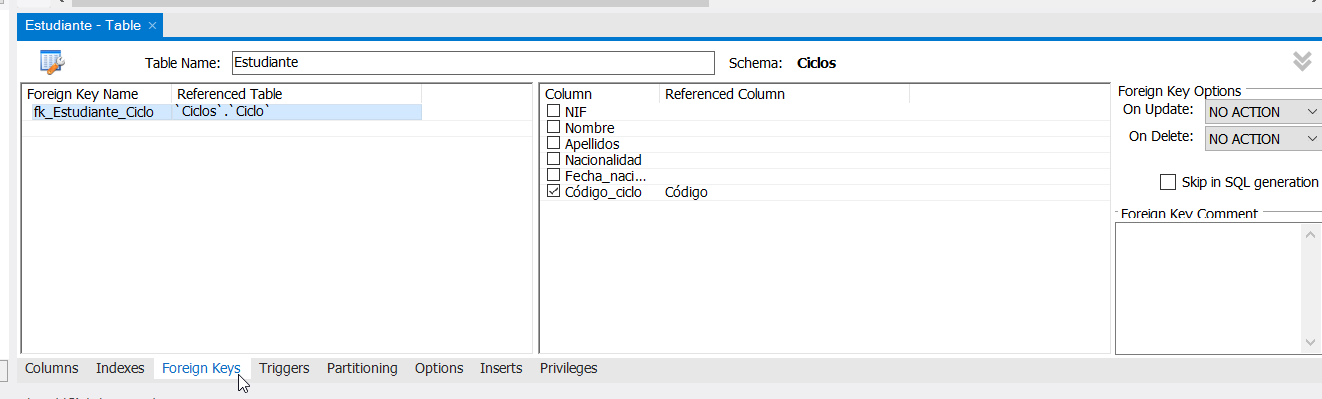
Ahora haz clic en las entidades Estudiante y Ciclo en ese orden, que es el sentido que indica la flecha del diagrama. Observa que ha aparecido un nuevo atributo en Estudiante, marcado en rojo (que indica clave ajena):



3.2.2. Haz doble clic en la tabla Estudiante y modifica el nombre que le ha dado de forma automática. Observa que ha sabido el tipo de datos necesario para la clave ajena, sencillamente poniendo el mismo que tenía la clave Código original en la tabla Ciclo.



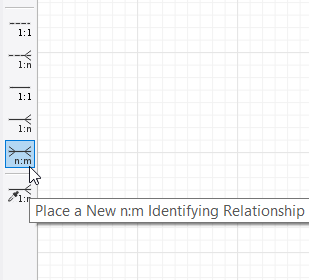
En la pestaña inferior “Foreign Keys” se puede ver la clave creada:



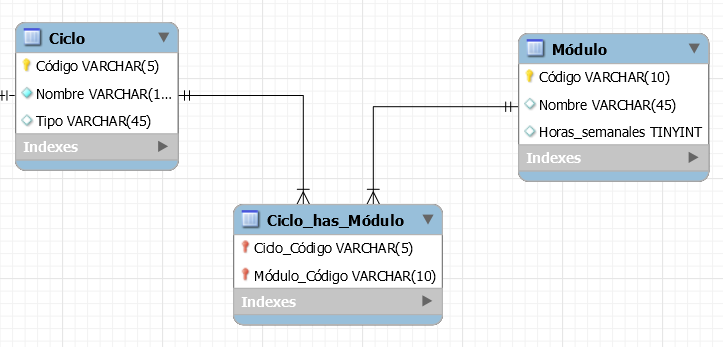
* El nombre de la clave (fK\_Estudiante\_Ciclo) es un nombre interno, porque todas las restricciones se almacenan en un nombre. Podría cambiarse pero no tiene efecto en nuestro problema.
* Observa la parte central: aparece marcada la columna “Código\_ciclo”, asociada a la “columna referenciada” “Código”.
* En la parte derecha se pueden añadir reglas de actualización o borrado en cascada. Se explorarán más adelante.

## 3.3. Modelado de la relación N:N “Pertenece a”

3.3.1. Utiliza el menú de relaciones para crear una relación N:N:



Haz clic en las tablas de Ciclo y Módulo y se creará la tabla correspondiente a esa nueva relación:



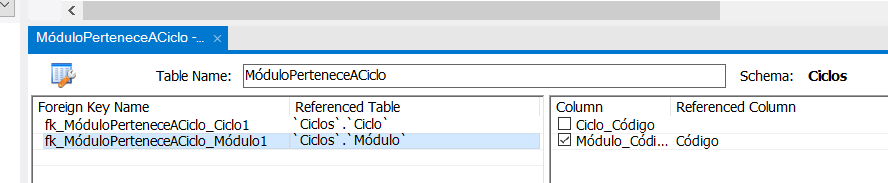
La llave roja indica “clave primaria y ajena”, ya que efectivamente estos campos son a la vez clave primaria de la tabla “Ciclo\_has\_Módulo” y ajena respecto a sus campos de origen.

**NOTA:** *Si no ves el icono de la llave roja, es por un fallo de ciertas versiones de MySQL Workbench en Windows. En la carpeta de la unidad tienes un documento “Arreglo MySQL Workbench.txt” que te explica cómo copiar la imagen correcta al sitio adecuado.*

Nota teórica sin efecto práctico: Observa que se considera “identifying” (línea continua), esto es así porque MySQL considera que existe dependencia en identificación entre Ciclo y la nueva tabla, y también entre Módulo y la nueva tabla, ya que esta nueva tabla no puede identificarse por sí misma (necesita las claves de ambas).

3.3.2. Renombra si lo deseas la tabla, para que se llame por ejemplo “MóduloPerteneceACiclo”, “Pertenece”, o como desees. También puedes renombrar los campos, que por defecto se nombran “Tabla\_Campo”, por ejemplo “Ciclo\_Código”.

Por último puedes examinar otra vez la pestaña de Foreign Keys:



Captura de MySQL Workbench a pantalla completa con tu diseño, que se vea la fecha del ordenador.

## 3.4. Proceso de Ingeniería directa para implementar la base de datos

3.4.1. Guarda el esquema para conservarlo.

3.4.2. Utiliza la opción Database 🡪 Forward Engineer para implementar la bases de datos en tu servidor.

## 3.5. Inserción de datos

3.4.1. Utiliza la vista de datos de MySQL Workbench para incluir datos en la base de datos. Puedes reutilizar los datos que utilizamos en clase para el ejemplo de ACCESS. Pon al menos cinco entradas en cada tabla.

Recuerda que debes cumplir las restricciones de clave ajena. Por ejemplo si a un estudiante le asignas un código de ciclo que no es válido,o obtendrás este error:

ERROR 1452: 1452: Cannot add or update a child row: a foreign key constraint fails

Es decir, “una restricción de clave ajena falla”.

Capturas de las cuatro tablas de datos una vez rellenas.

# Nota final

Existen más formas de hacer las cosas en MySQL. Por ejemplo:

### Creación manual de una relación N:N

La tabla de la relación N:N podría crearse manualmente y establecer relaciones 1:N por separado con las dos tablas:

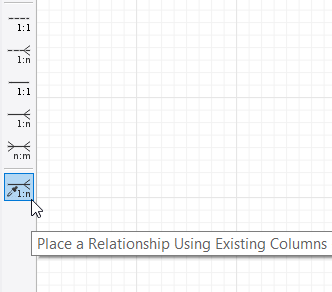
* Crea manualmente una tabla vacía “MóduloPerteneceACiclo”
* Añade una relación 1:N desde “MóduloPerteneceACiclo” hacia “Ciclo”
* Añade una relación 1:N desde “MóduloPerteneceACiclo” hacia “Módulo)
* Marca los dos nuevos campos como claves primarias

El efecto será el mismo.

### Creación manual de una relación 1:N sobre un campo ya creado

A veces ocurre que al crear la tabla en MySQL Workbench ya hemos creado el campo que será clave ajena, por ejemplo “Código\_ciclo” en la tabla Estudiante. El botón de relación 1:N, sin embargo, no va a utilizar tu campo sino que va a crear uno nuevo.

Si quieres reutilizar un campo ya existente para que sea clave ajena, debes usar este botón:



Ahora haz clic en el campo que será clave ajena (por ejemplo “Código\_ciclo” y después en el campo referenciado (“Código” en la tabla “Ciclo”). El resultado será el mismo.