

## Licenciatura en Sistemas - Introducción a Bases de Datos

Titular: Ing. Federico Ribeiro  
Ayudante: Lic. Leandro Ríos

[fribeirounla@gmail.com](mailto:fribeirounla@gmail.com)  
[leandro.rios.unla@gmail.com](mailto:leandro.rios.unla@gmail.com)



### Modelo Entidad - Relación. Diagrama Entidad Relación.

Continuando con los distintos formalismos de los que disponemos para expresar nuestros diseños de datos. En esta oportunidad veremos el diagrama entidad-relación (DER o ERD por su sigla en inglés)

El diagrama entidad relación es similar al diagrama conceptual en que nos permite plasmar la estructura del modelo de datos que estamos diseñando. Así, de manera similar, tendremos entidades con atributos y relaciones entre las mismas, con sus cardinalidades y modalidades correspondientes.

La ventaja de emplear DER en lugar de DC es que nos deja con un diagrama más cercano a la implementación final de nuestra base de datos. Debido a que el motor de bases de datos relacional representa entidades y relaciones empleando tablas, tendremos una correlación 1:1 entre las entidades del DER y las tablas de la base de datos. Esto, sin embargo, también es una desventaja para comprender el modelo de datos: muchas de las relaciones, especialmente las n:n (muchos a muchos) se implementan también con tablas y por ello terminan “transformándose” en entidades, que pueden o no tener significado en el universo de discurso o minimundo (el universo de discurso es el conjunto de cosas de las que hablamos para definir el problema a resolver, en el caso de una aplicación de gestión comercial podríamos hablar de clientes, proveedores, pedidos, etc.). El DC deja esto en claro, ya que la representación de las relaciones es explícita y distinta a las de las entidades. Entonces, el DC es una representación más fiel de la estructura de nuestro modelo de datos, en términos de entidades y relaciones, mientras que el DER nos ofrece una representación más cercana a la implementación en la base de datos.

Pueden encontrar en algunos textos que se denomina DER tanto al DC como al DER, en este curso haremos la distinción entre los dos tipos de diagramas.

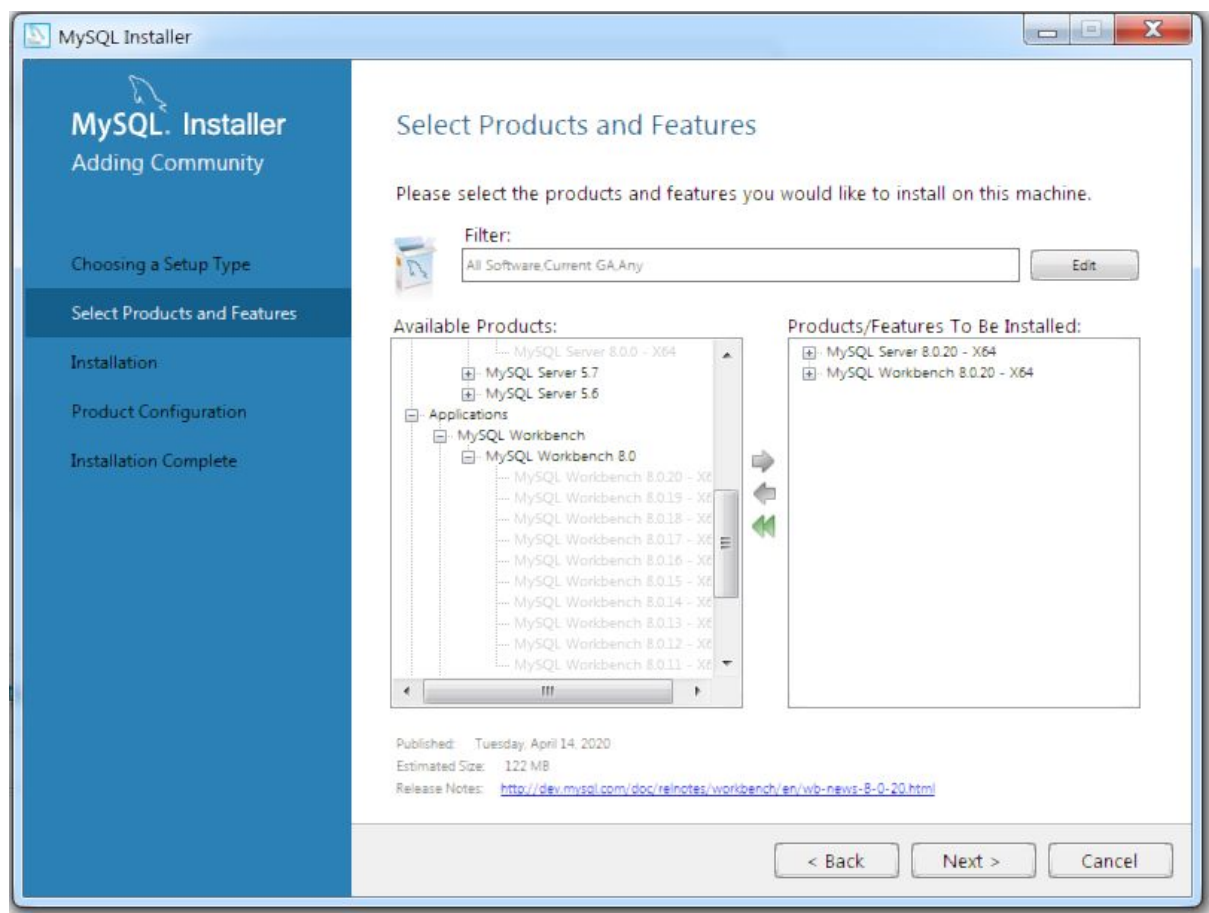
A fin de realizar los diagramas entidad-relación emplearemos la herramienta de diseño del servidor de base de datos MySQL que utilizaremos durante la cursada: MySQL Workbench. MySQL es un servidor de bases de datos de código abierto con licencia GPL en su versión Community, la versión Enterprise ofrece funcionalidades desarrolladas por Oracle de código propietario. Tiene amplia difusión en el mercado y está disponible gratuitamente.

Como ya hemos visto, hay dos componentes en un sistema de bases de datos: el servidor, que almacena, organiza y garantiza la seguridad y consistencia de los datos, y el cliente, que es el que hace uso de dichos datos. A los fines de las prácticas que realizaremos,

instalaremos el servidor (MySQL server) y el cliente (MySQL Workbench) en la misma máquina, pero nada impide, y de hecho es muy usual, que ambos se encuentren en distintas máquinas conectados por medio de una red de alguna clase. De la misma manera, pueden existir distintos clientes corriendo en distintas máquinas conectados al mismo servidor. De hecho, esta última es la configuración más usual.

Deberemos descargar e instalar entonces éstos dos componentes. A tal fin, nos dirigiremos a la URI <https://dev.mysql.com/downloads/installer/> y descargaremos la versión installer-web-community (un archivo .msi). En la opción figura que es para 32 bits, pero esa es la arquitectura en la que corre el instalador, cuando lo ejecutemos podremos instalar el servidor y cliente para x64. En la página siguiente, haremos click en la frase “No thanks, just start my download.” para comenzar la descarga.

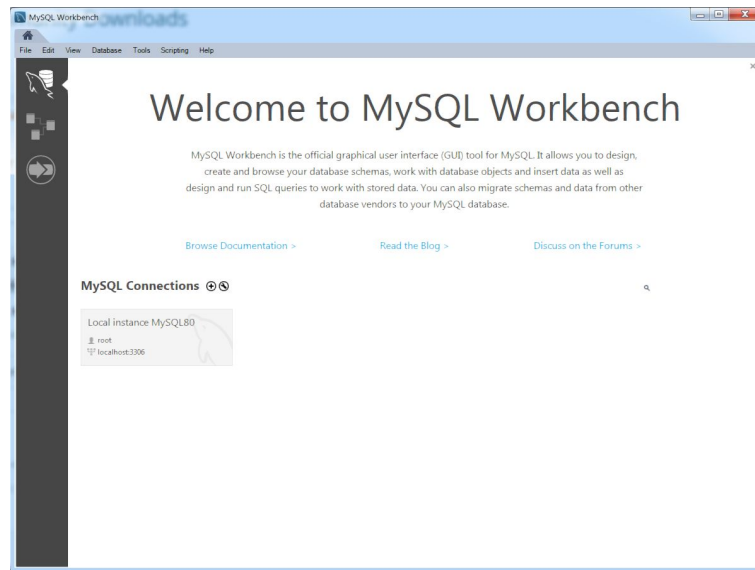
Una vez finalizada la misma, ejecutaremos el archivo descargado y en la pantalla inicial elegiremos la opción “Custom” En la siguiente ventana elegiremos para instalar el servidor más moderno disponible para nuestra arquitectura (al momento de la escritura es 8.0.20) y la versión más reciente de MySQL Workbench, que encontraremos dentro del apartado “Applications”. Debe quedar de la siguiente manera:



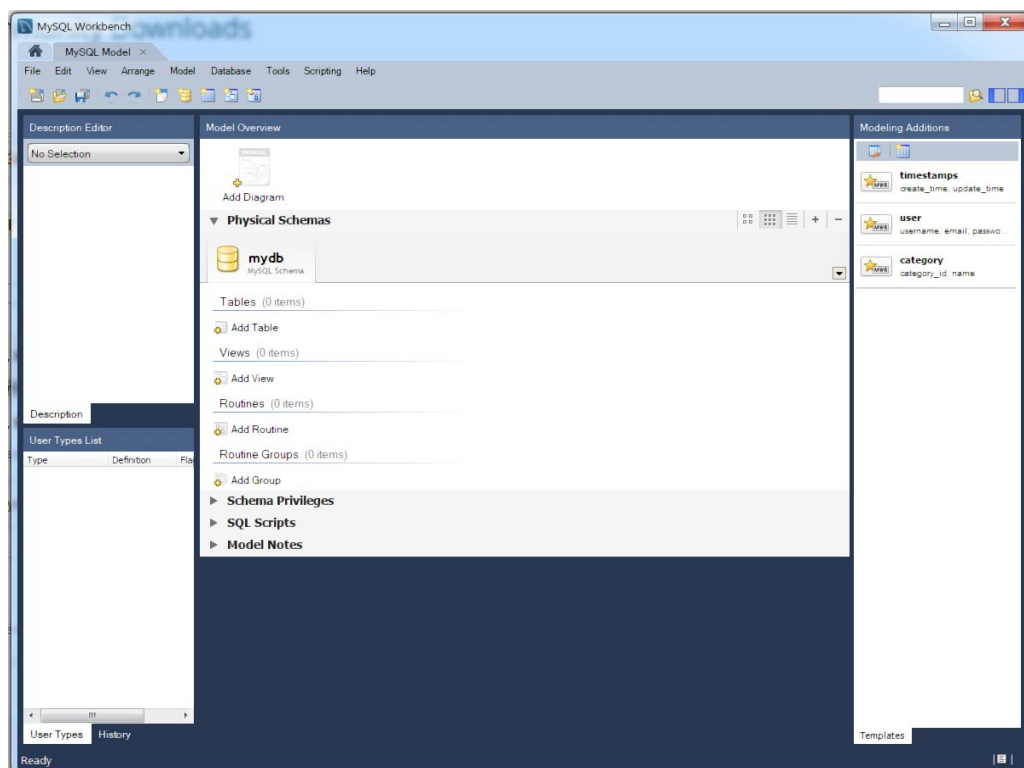
Click sobre “Next” y luego sobre “Execute”; comenzará la descarga e instalación del software. Cuando se haya completado, daremos click en “Next” para comenzar con la configuración del servidor. En la pantalla siguiente seleccionaremos la opción “Standalone MySQL Server / Classic MySQL Replication” y click sobre “Next”. En la pantalla siguiente dejaremos las opciones como las ofrece y click sobre “Next” hasta llegar al apartado

“Accounts and Roles”. Allí ingresaremos un password para la cuenta root. El usuario root es el que tiene la máxima autoridad sobre la base de datos, puede agregar y eliminar usuarios, crear y eliminar esquemas y en general, configurar el servidor de bases de datos en todos sus aspectos.

A continuación, hacemos click sobre “Next” hasta que el botón se convierta en “Execute”. Click sobre el mismo y aguardamos hasta que termine la configuración. Click sobre Finish y si todo anduvo bien, debería abrirse la ventana de MySQL Workbench:

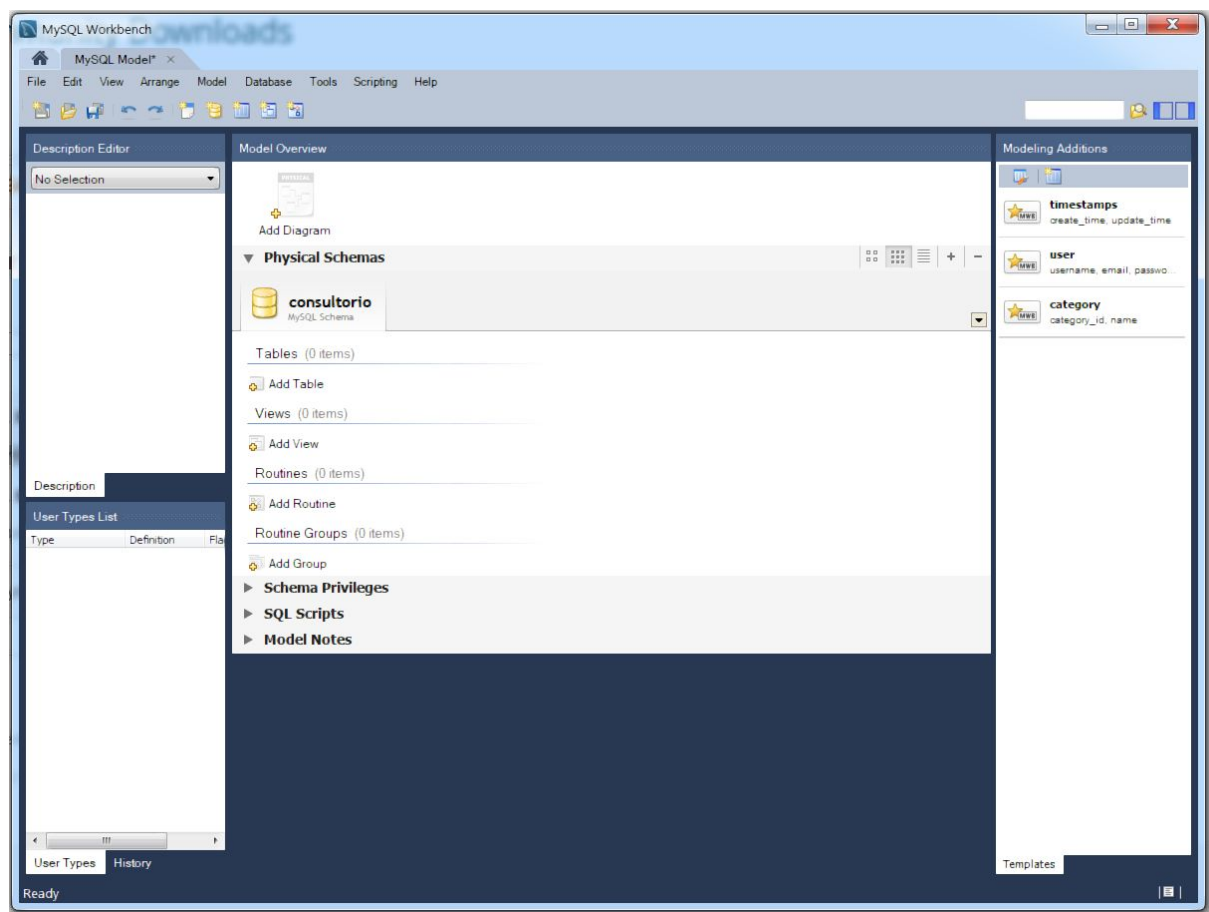


Vamos a comenzar, entonces a realizar nuestro DER. Para ello, vamos file/New Model. Veremos lo siguiente:



MySQL Workbench nos ofrece un nuevo esquema (schema en inglés) en el cual plasmar nuestro modelo. Un esquema de base de datos es el conjunto de tipos y estructuras de datos que hacen a la solución de un problema. En nuestro caso, una vez que volquemos nuestro diseño del modelo del consultorio a la base de datos, lo haremos sobre un conjunto de tablas, campo, índice, etc. Todo ello conforma el esquema “consultorio”. En un servidor de bases de datos pueden coexistir más de un esquema, por lo que un esquema no es sinónimo de base de datos. Una base de datos suele ser un conjunto de esquemas, relacionados o no. En MySQL esto se presta a confusión, ya que como veremos más adelante emplea los términos “schema” y “database” como sinónimos. Cuando llegue el momento, volveremos a advertirlo.

Hacemos click derecho sobre “Mydb” (el nombre que Workbench asigna a un esquema por defecto) y seleccionamos la opción “Edit Schema”. En el campo “Name”, cambiamos el nombre por “consultorio” y cerramos la pestaña. Debe quedar de la siguiente forma:



A continuación, doble click sobre “Add diagram”. Workbench nos responderá creando un área de dibujo en la que podremos crear nuestro diagrama.

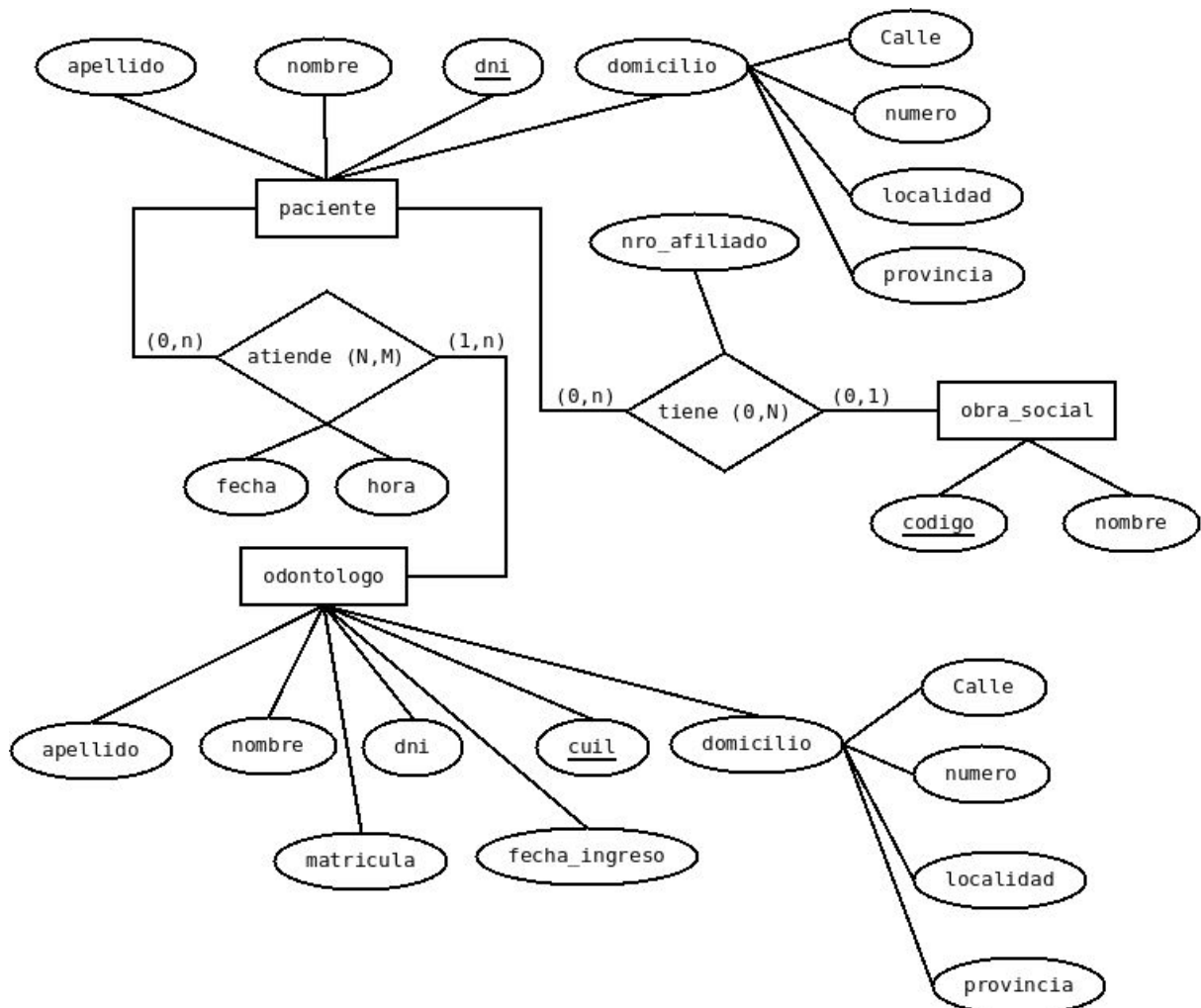
A la izquierda del área de dibujo encontramos las herramientas necesarias: por ahora emplearemos new table (nueva tabla), y todas las dedicadas a relaciones (relationship).

Comenzaremos haciendo el diagrama del enunciado 1 (consultorio), que reproducimos a continuación:

### Enunciado 1

Para informatizar la operatoria de una clínica dental se realizó el siguiente relevamiento: La clínica lleva registro de los odontólogos que allí atienden. De cada uno de ellos se conoce su apellido, nombre, DNI, CUIL, fecha de ingreso, domicilio(calle, número, localidad y provincia) y matrícula. De los pacientes se registra su apellido, nombre, DNI, domicilio (calle, número, localidad, provincia), obra social si la tiene (nombre de la obra social, código y número de afiliado). Cada vez que se otorga un turno se registra el nombre y apellido del paciente, los del odontólogo y la fecha y hora del turno.

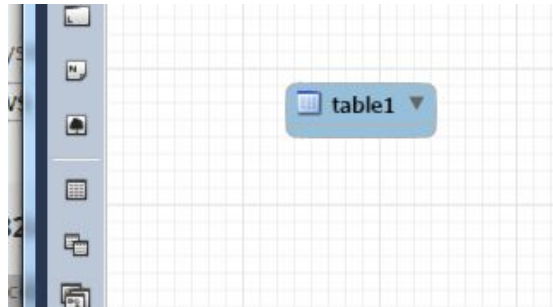
Tomaremos como base para la creación del DER el DC que realizamos previamente.



Comenzaremos por crear las entidades en nuestro nuevo DER: para ello, elegimos la herramienta "Place a new table" o presionamos la letra T:

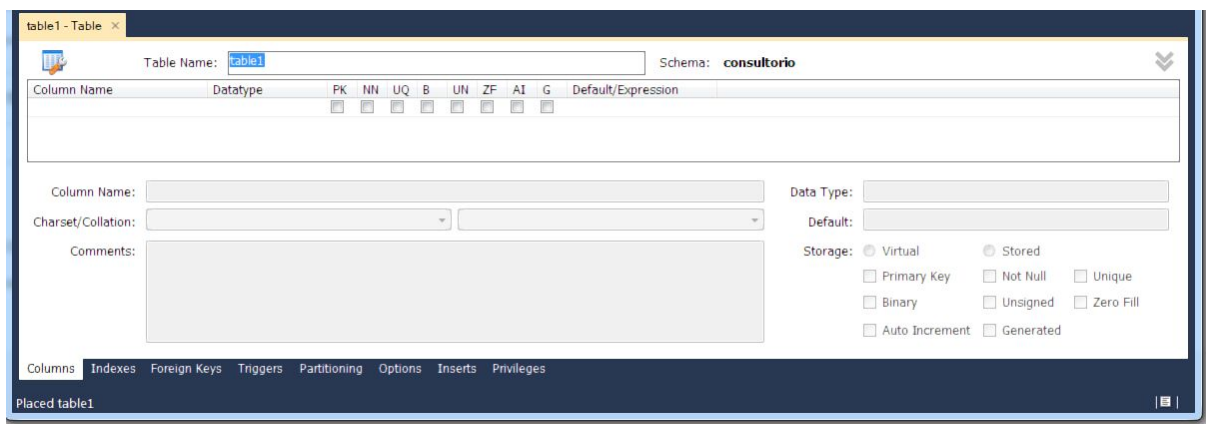


y hacemos click en el lugar del diagrama donde queramos ubicarla. Workbench colocará nuestra entidad con el nombre por defecto “table1”:



La herramienta de generación de diagramas de Workbench tiene algunas particularidades que la diferencian de un diagrama E-R convencional. Por ahora, vemos a que a las entidades les pone el nombre de “tabla”. En este caso, tabla1 (table1 en inglés). Esto es porque la herramienta está orientada al diseño de bases de datos para un motor relacional en el que todo, salvo los atributos, se representa por medio de tablas.

Comencemos a darle cuerpo a nuestra entidad: doble click sobre la misma y se abrirá el área de edición de la misma:



En primer lugar, vamos a darle un nombre adecuado a esta entidad. En el campo “Table name”, ingresamos el nombre de la entidad, en este caso, paciente.

Vamos a agregar atributos a esta entidad. Cada atributo se ingresa como una fila de la tabla que aparece inmediatamente debajo. En la columna denominada “column name” ingresaremos el nombre del mismo, en este caso, dni.

En el campo siguiente podremos ingresar el tipo de datos con el que representaremos este atributo. En este caso utilizaremos el tipo INT (entero de 4 bytes con signo).

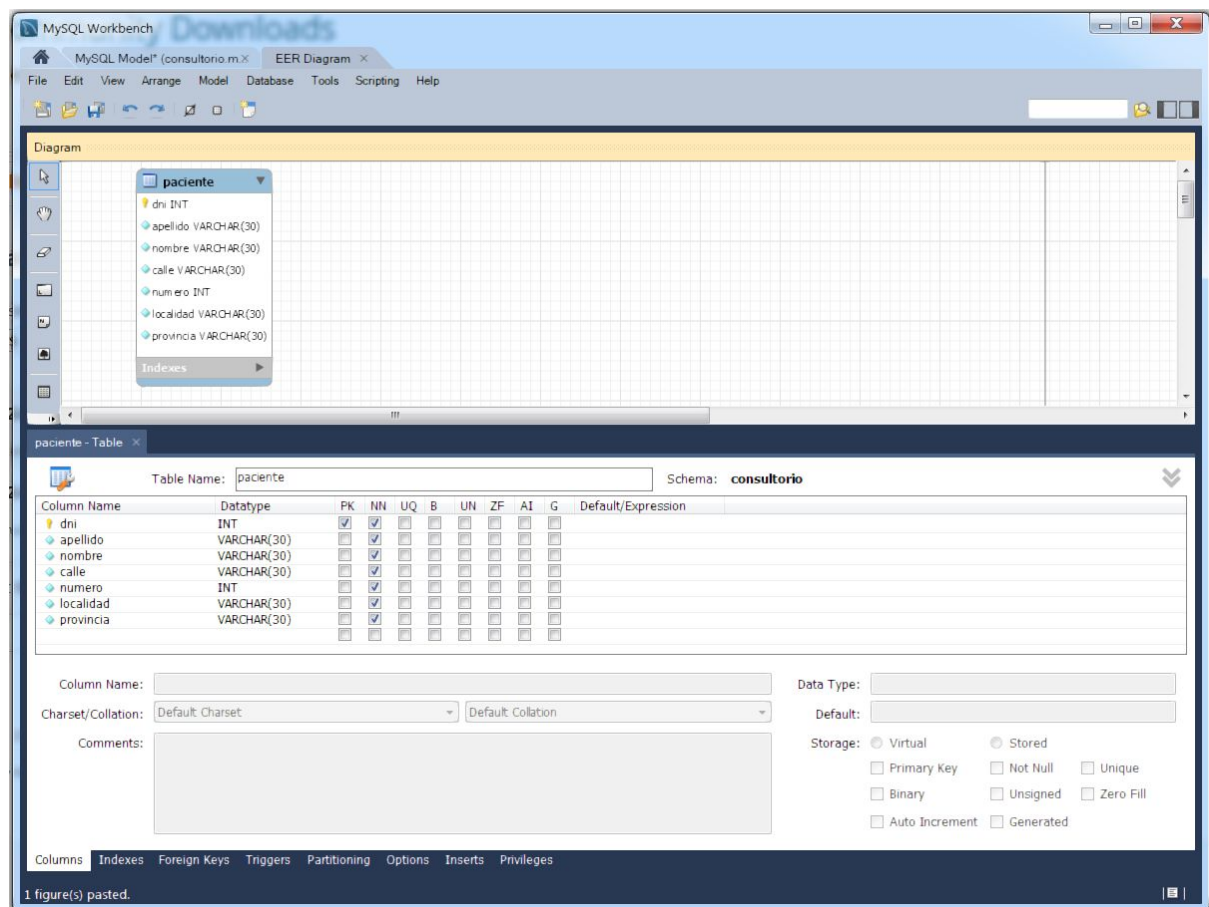
Workbench marca el primer atributo que ingresamos como clave primaria, y por ello los campos PK (primary key, clave primaria) y NN (not null, no nulo). La restricción not null se aplica siempre para la clave primaria. Si Workbench no lo hubiera marcado, marcaríamos PK. NN se marcará automáticamente.

Continuamos con el resto de los atributos, hacemos click en la fila siguiente y agregamos el atributo apellido. En este caso el tipo es VARCHAR. VARCHAR es una cadena de



caracteres de tamaño variable. El número 45 entre paréntesis es la longitud máxima (por defecto) del string que podremos almacenar en el campo. Así, MySQL sólo empleará la cantidad de espacio necesaria para almacenar el string que estemos ingresando. Si el apellido fuera “López” sólo almacenaría un string de 5 caracteres. Si precisamos almacenar strings de mayor tamaño, cambiaremos el 45 a un número mayor, con un máximo de 65535. En nuestro caso, 30 caracteres serán suficientes, por lo que cambiaremos la longitud a dicho valor.

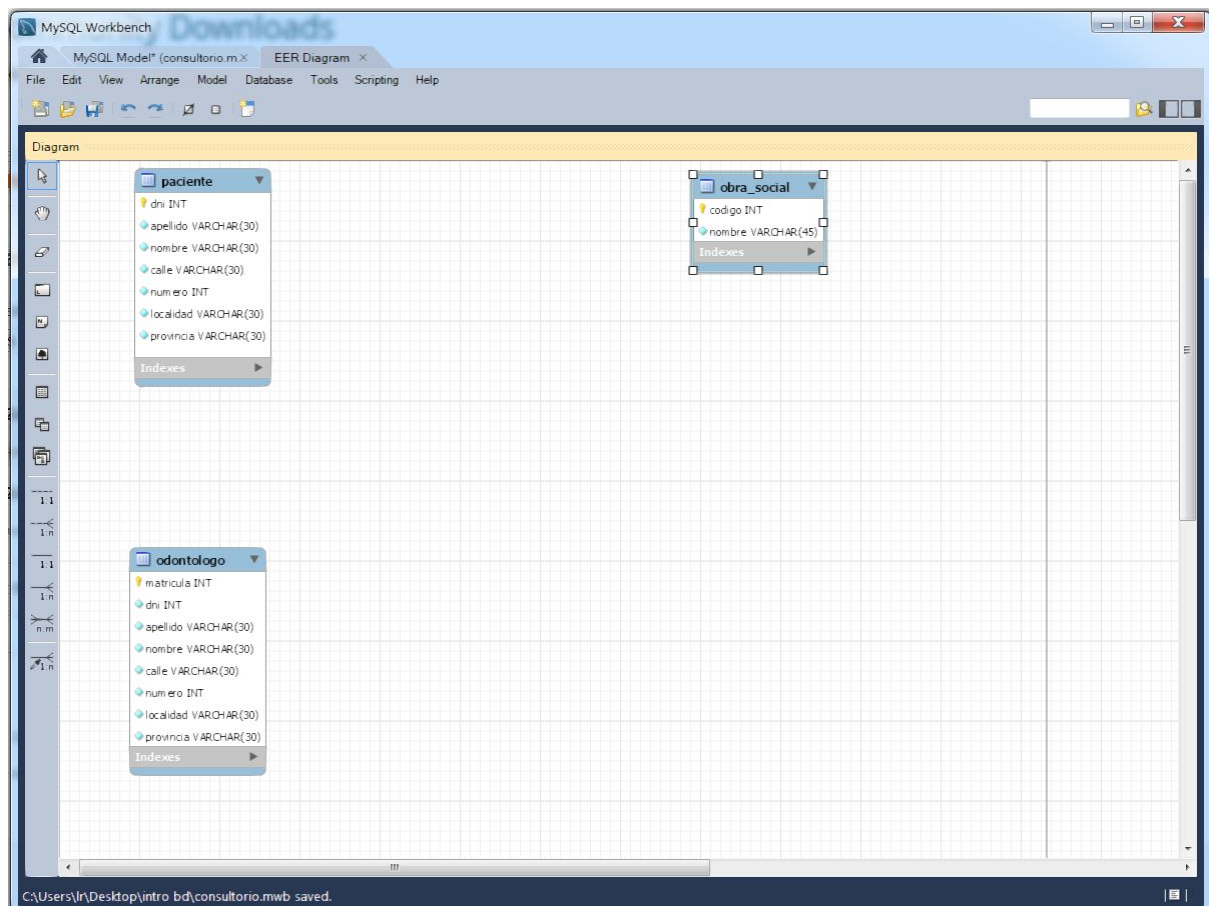
El resto de los atributos se agregan de la misma manera. El atributo compuesto domicilio se agrega como cada uno de sus componentes por separado. Si algún atributo se agregó de más, daremos botón derecho sobre el mismo y elegiremos la opción “Delete selected”. Finalmente debe quedar así:



A todos los atributos les hemos indicado la restricción not null. Esto implica que dicho atributo no podrá ser nulo, es decir, que no se puede dejar sin asignar, o lo que es lo mismo, que su presencia es obligatoria. De esta manera nos aseguraremos de que no se pueda ingresar un paciente sin apellido, por ejemplo. Este es otro caso en que el DER realizado en Workbench difiere del DER tradicional, ya que las restricciones no suelen indicarse en el mismo.

Podemos cerrar la pestaña de edición de la entidad para ampliar el área de diagramado. Continuamos agregando el resto de las entidades de la misma manera. Para el atributo fecha de ingreso de la entidad odontólogo emplearemos el tipo DATE.

El diagrama debe quedar de esta manera:



## Relaciones

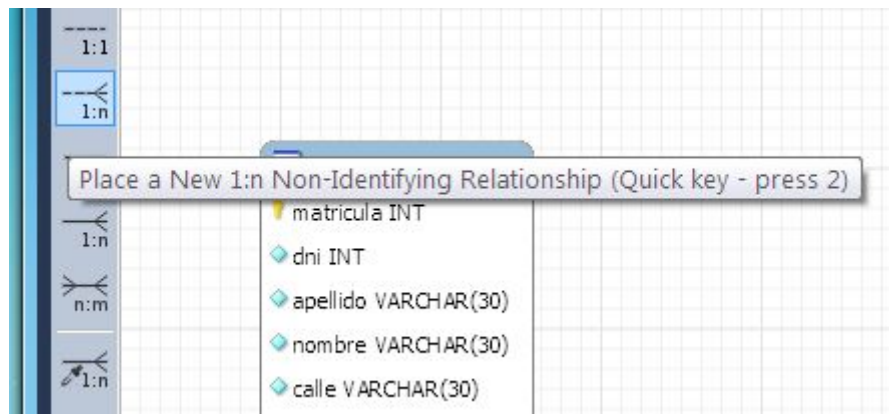
Vamos a agregar las relaciones a nuestro diagrama. Mysql Workbench emplea la notación Bachman o “Pata de gallo” (crow’s foot). En esta notación las cardinalidades parciales se representan como cardinalidad y modalidad, por medio de líneas. Ya no empleamos números para indicar cardinalidades, a menos que queramos indicar el límite superior de una relación con orden n.

Se denomina modalidad a la cardinalidad parcial mínima de una relación. Esto quiere decir que, si tenemos una cardinalidad parcial (0,n), siendo 0 la cardinalidad parcial mínima, la relación puede existir como no. En el ejemplo del consultorio, la relación “tiene” entre paciente y obra social tiene una cardinalidad de (0,n) del lado de la izquierda, lo que indica que una obra social puede tener o no pacientes afiliados. Si tiene 0 pacientes, la relación no existe y por ello decimos que su modalidad es opcional o no obligatoria. Puede existir como no.

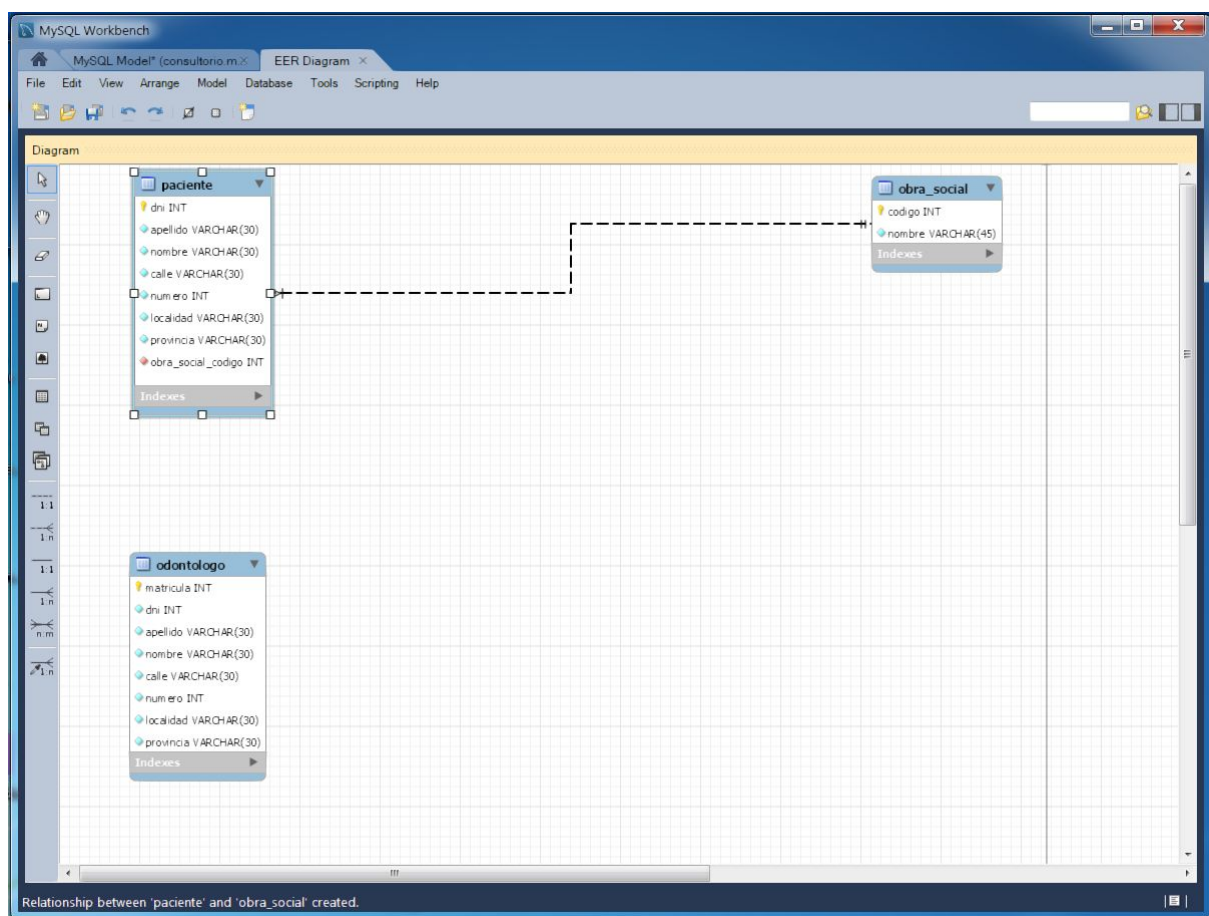
En cambio, la relación atiende entre odontólogo y paciente tiene una cardinalidad parcial de (1,n) del lado inferior. Esto es así porque para que un paciente sea atendido por un odontólogo, es obligatorio que un odontólogo participe en la relación. Como mínimo tiene que haber uno, luego el paciente podrá atenderse con otros si lo considera necesario. En este caso decimos que la modalidad de la relación es obligatoria.



Comenzaremos creando la relación “tiene” entre paciente y obra social. A tal fin, emplearemos la herramienta de creación de relaciones 1:n no identificatoria o presionamos la tecla 2:



A continuación, hacemos click primero sobre la entidad del lado de “muchos” de la relación (en este caso,, paciente) y luego en la del lado de 1 (obra social). Debe quedar así:



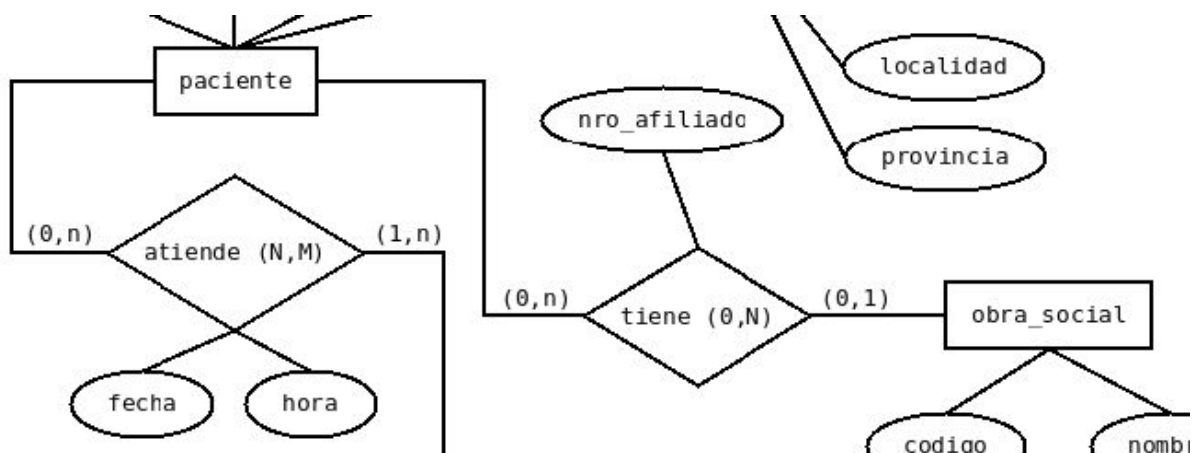
La línea punteada indica que se trata de un relación “no identificatoria”. Esto significa que ninguna de las entidades participantes es débil. Si una de ambas entidades fuera débil, la

relación sería identificatoria, ya que la clave primaria de la entidad fuerte se emplearía para identificar a la débil.

Veamos ahora qué significan las líneas en los extremos de la relación. Del lado de paciente:



La forma en que termina la línea punteada, con las tres líneas que semejan la pata de un ave es por lo que se denomina “pata de gallo” a esta notación, y esas tres líneas indican que la cardinalidad parcial máxima de la relación es “muchos” (n). La línea vertical inmediatamente antes, indica la modalidad obligatoria de la misma, o lo que es lo mismo, que su cardinalidad parcial mínima es 1. Pero un momento: Dijimos que la cardinalidad parcial mínima de esta relación es 0, o lo que es lo mismo, que la relación es opcional:



Lo que ocurre es que aún nos falta terminar de definir la relación. Si miramos con atención, veremos que Workbench acaba de agregar un nuevo atributo o campo a la entidad paciente: obra\_social\_codigo. Este nuevo atributo se agrega para que funciones como clave foránea de obra social en paciente. De este modo, teniendo el paciente, podremos mirar cuál es su obra social mirando dicha clave foránea. Si necesitamos conocer los detalles de la obra social, con dicho código haremos una búsqueda por la misma, incluso existen formas de hacer ambas consultas simultáneamente; esto lo veremos cuando comencemos a estudiar el lenguaje SQL.

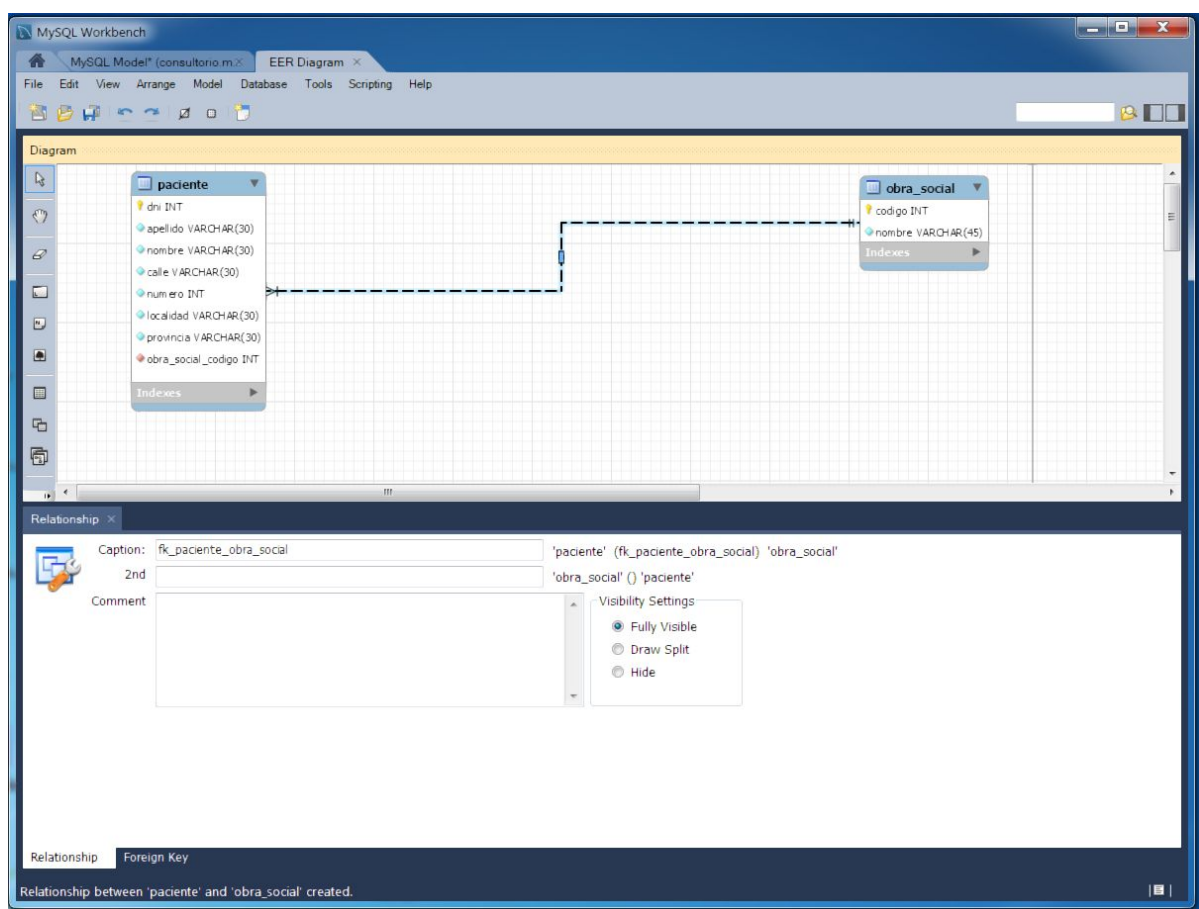
El problema es que así como está definida, la existencia de la clave foránea es obligatoria. Y del otro lado ocurre lo mismo: en el extremo de obra\_social de la relación podemos ver lo siguiente:



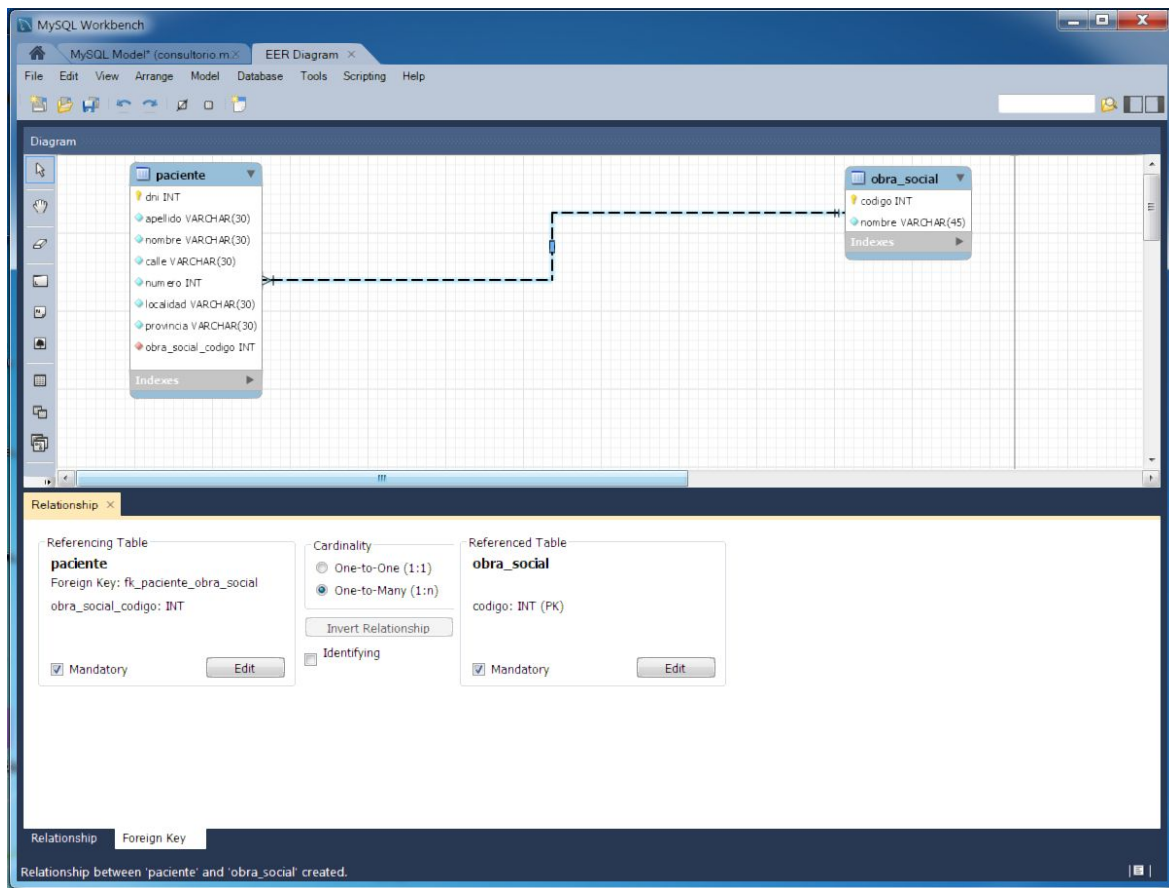
En este caso, vemos dos líneas verticales. La de la derecha (la más cercana a la entidad) indica la cardinalidad parcial máxima. Una línea vertical indica que la cardinalidad es 1. Si fuera n, tendríamos la “pata de gallo” como en el caso anterior. Esto coincide con lo que vemos en el DC del problema.

La línea de la izquierda, en cambio (la más alejada de la entidad) nos indica modalidad obligatoria o cardinalidad mínima parcial 1. Igual que en el caso anterior, esto no está bien, ya que la cardinalidad mínima parcial es cero, o lo que es lo mismo, su modalidad es opcional.

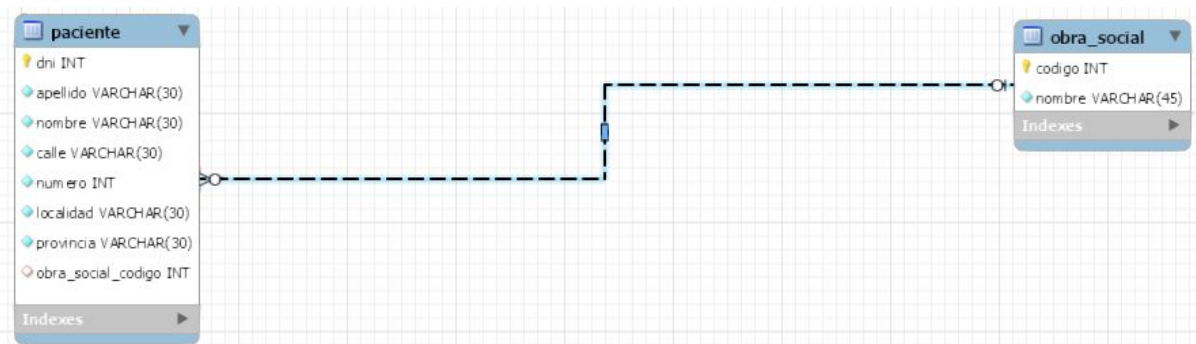
A fin de indicar que la relación es opcional, haremos doble click en algún lugar de la línea de la relación para poder editarla. Debemos ver lo siguiente:



Haremos click abajo, en la pestaña “Foreign key” (clave foránea) a fin de editarla. deberemos ver lo siguiente:



Quitaremos el tilde del check “Mandatory” (obligatoria) en ambos lados de la relación y cerraremos la pestaña “relationship” (relación). La relación debe quedar de la siguiente manera:



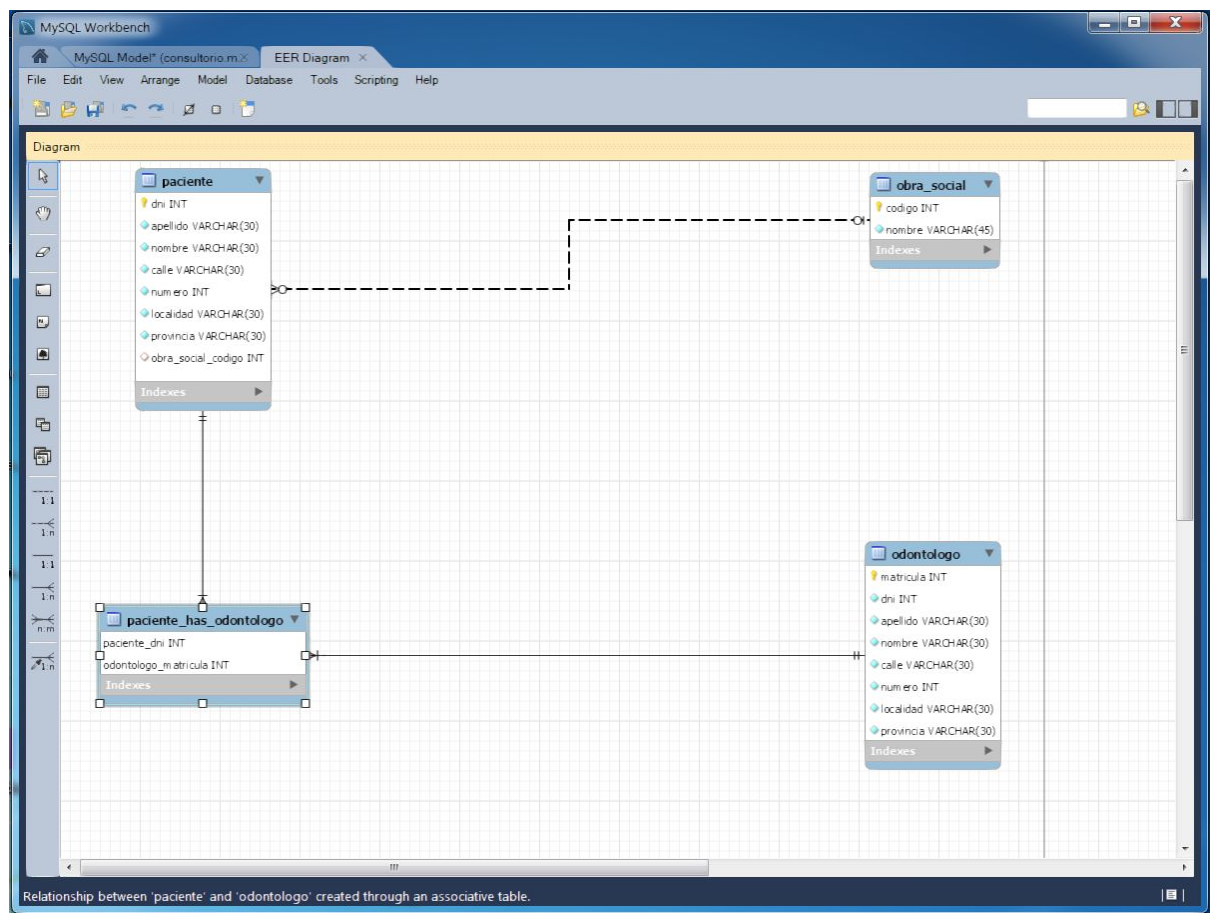
Vemos como la línea de modalidad cambió a un círculo, esto indica que es opcional o no obligatoria.

Entonces, recapitulando, la “pata de gallo” del lado de paciente nos indica una cardinalidad parcial máxima de n, y corresponde con la n de la cardinalidad parcial de la relación: (0,n). El círculo (como ayuda memoria podemos pensarlo como un 0) indica que la cardinalidad parcial mínima es 0 y corresponde con el 0 de la cardinalidad parcial de la relación: (0,n). También decimos que indica que la modalidad de la relación es opcional.

Habiendo terminado con la relación tiene entre paciente y obra social, continuaremos con la relación “atiende” entre paciente y odontólogo. Emplearemos la herramienta de creación de relaciones n:n identificatoria o presionamos la tecla 5:

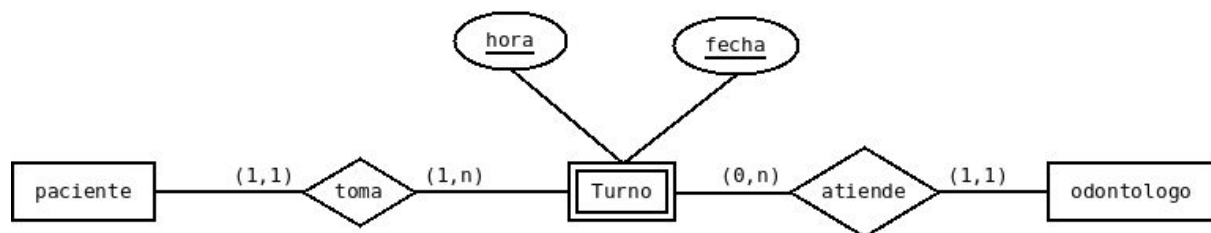


Haremos click sobre una de las entidades (cualquiera, no importa) y luego sobre la otra. Debe quedar así:



A fin de que el diagrama quede más claro, corrimos la entidad “odontólogo” hacia la derecha. Pero, ¿Qué ocurrió? ¿Tenemos una nueva entidad? Si y no. El problema es que

debido a que MySQL utiliza tablas para representar las relaciones y las entidades, lo que Workbench nos está mostrando es la representación de la relación como una tabla. De todos modos, pensemos un poco en nuestro diseño: ¿No podemos tener una entidad llamada “turno” que almacena la información de un turno, y que esté relacionada con paciente y odontólogo? Si la respuesta es sí, entonces esta nueva entidad tiene algún significado. Volvamos al DC y agreguemos esta nueva entidad:



Hemos obviado los atributos de paciente y odontólogo para mayor claridad. Esto que tenemos aquí es muy parecido a lo que acaba de hacer Workbench. Esto es porque hay una cierta dualidad entre las relaciones y las entidades, y muchas relaciones pueden transformarse en entidades; esto es cierto para las relaciones n:n. También es cierto para las relaciones de orden mayor a 2 (con más de dos entidades participantes).

La entidad turno es débil porque no tiene una clave primaria propia: la misma se forma por medio de las claves foráneas de paciente y odontólogo y con fecha y hora. Esto es porque sólo puede haber un turno de un paciente para un odontólogo en una fecha y hora determinadas. Tenemos entonces, una entidad débil con una clave compuesta por dos claves foráneas y dos atributos propios, que funcionan como discriminantes.

Esto podemos verlo así porque, en realidad, en el modelo relacional todas las cosas, salvo los atributos, son relaciones. Una entidad es un conjunto de atributos relacionados con una clave. Decimos que la clave “determina” (una relación) a los atributos. O al revés, decimos que los atributos “dependen” de la clave (otra relación). El establecimiento de entidades es de alguna manera arbitrario y responde al universo del discurso y a la mejor forma de solucionar nuestro problema. Es por esto que siempre hacemos hincapié en que las soluciones no son únicas, y mucho menos las representaciones formales. Es importante que puedan desarrollar esta forma de ver las cosas desde distintos puntos de vista, para poder aplicar el más conveniente para resolver el problema en cuestión.

Volviendo a nuestro diagrama, lo primero que podemos ver es que la nueva entidad se llama “paciente\_has\_odontologo”, este es un nombre que ha asignado Workbench por defecto. Vamos a cambiarlo por algo con más sentido para nosotros. Siguiendo el diagrama recién visto, vamos a ponerle el nombre “turno”. Para ello la editamos a como a cualquier entidad: doble click sobre la misma y se abrirá la ventana de edición.

Otra cosa que será necesario modificar es que a la entidad turno le faltan los atributos fecha y hora. Se agregan de la misma manera que hemos hecho hasta ahora con las otras entidades del modelo. Recuerden que deben marcarse como clave primaria. Para indicar una clave primaria compuesta, se deben indicar cada uno de sus atributos como clave primaria.



Conviene tener presente que cada vez nos acercamos más al nivel de la implementación: Luego de esta fase del diseño, sólo queda crear el modelo en la base de datos. Veremos cuando lo hagamos que la correspondencia entre el DER que estamos realizando y la implementación será completa.

Lo que nos queda modificar son las cardinalidades. Con lo visto hasta ahora, deberían ser capaces de hacerlo por ustedes mismos. Les dejamos la versión final con fines de comprobación:



Algunas observaciones:

Las líneas de las relaciones entre turno y odontólogo y paciente son llenas (no punteadas) porque se trata de una relación identificatoria.

En el DER tradicional (el que verán o vieron en Ingeniería del Software 1), pueden ponerse los nombres de las relaciones en las líneas de las mismas. En esta cátedra no nombramos las relaciones en el DER, sí lo hacemos en el DC.