



Tutorial de MySQL Workbench

Esquema relacional

Lic. Mariana Adó



Este documento está disponible en acceso abierto bajo la licencia
Atribución-NoComercial-CompartirlGual 2.5 Argentina (CC-BY-NC-SA 2.5 AR DEED)
(<https://creativecommons.org/licenses/by-nc-sa/2.5/ar/>)
Primera edición: 2023, Buenos Aires, Argentina.

Autora: Mariana Adó.



Índice

¿Qué es MySQL WorkBench?	5
¿Cómo obtener MySQL WorkBench?	5
¿Para qué sirve la herramienta?	5
Tipo de notación de este tutorial	6
Notación para los objetos y relaciones.....	6
Notación para la cardinalidad.....	6
Esquema relacional	7
Crear el diagrama.....	7
Cambiar el nombre del esquema de la base de datos.....	11
Crear una tabla en el esquema relacional.....	12
Crear las columnas de una tabla.....	14
Propiedades de los columnas.....	15
Claves de la tabla.....	16
Clave Primaria.....	16
Clave Primaria Subrogada.....	17
Claves primarias compuestas.....	18
Claves Candidatas (o secundarias).....	19
Claves candidatas simples.....	19
Claves candidatas compuestas.....	19
Columnas de la tabla y tipos de datos.....	21
Dominios definidos por el usuario.....	22
Relaciones entre las tablas.....	23
Tipos de relaciones.....	23
Relaciones con conectividad N:M.....	24
Claves Foráneas.....	25
Índices automáticos para FK.....	27
Usando PK subrogada para las relaciones N:M.....	28
Relaciones con conectividad 1:N.....	30



Propagación de clave.....	30
Cardinalidad (1,1) → (1, N) o (1,1) → (0, N).....	30
Cardinalidad (0,1) → (1, N) o (0,1) → (0, N).....	32
Creando una nueva tabla.....	34
Cardinalidad (0,1) → (1, N) o (0,1) → (0, N).....	34
Relaciones con conectividad 1:1.....	35
Relaciones Mandatorias y No Mandatorias.....	35
Etiquetar las relaciones.....	37
Jerarquías (“Es_un”) - Relaciones 1:1 Identificables.....	38
Relaciones Recursivas.....	41
Entidades débiles por identificación.....	43
Agregaciones.....	47
Relaciones Ternarias.....	51
Historial.....	54
Esquema Físico.....	57
Ingeniería hacia adelante.....	57
Visualizar el esquema físico de la base de datos.....	67
Anexo.....	69
Caso de Estudio Médicos sin Fronteras.....	69



¿Qué es MySQL WorkBench?

MySQL Workbench es un entorno gráfico para el diseño y documentación de Bases de Datos (BD) relacionales, pensada para ser usada con el Sistema de Gestión de Bases de Datos (SGBD) MySQL Server.

Existen dos versiones del producto, una es Open Source (OS) y la otra es una versión comercial. La versión comercial proporciona algunas funcionalidades que pueden ser de interés en algún ámbito particular, aunque la versión OS resulta suficiente para la realización de las prácticas de las asignaturas en la universidad.

Aprende más sobre MySQL WorkBench en el siguiente link:

<https://www.mysql.com/products/workbench>

¿Cómo obtener MySQL WorkBench?

Los links de descarga son los siguientes:

Windows y Linux:

<https://dev.mysql.com/downloads/workbench>

Para la elaboración del tutorial se utiliza la versión 8.0.32 Community Edition (CE).

¿Para qué sirve la herramienta?

La herramienta se utiliza para representar el diagrama del esquema relacional de una base de datos, es decir, las tablas y sus columnas, los tipos de datos de las columnas (dominio de los atributos), las claves primarias, claves candidatas, las claves subrogadas, las claves foráneas y a quienes referencian.

Por último, usando ingeniería hacia adelante, se crea el modelo físico en el SGBD MySQL Server 5.5 o versiones posteriores.



Tipo de notación de este tutorial

Notación para los objetos y relaciones

MySQL WorkBench permite utilizar distintas notaciones tanto para los objetos como para las relaciones del esquema relacional. En este tutorial se utiliza la notación WorkBench (Default) para los objetos y Crow's Foot (IE) para las relaciones. Y es la que usaremos en este presente tutorial.

Notación para la cardinalidad

En el esquema conceptual, la cardinalidad se denota como un par ordenado para cada entidad que participa en la relación: (card mínima, card máxima)

Existen dos formas de hacerlo:

- **Look Here:** representamos la cardinalidad de la entidad en la relación, ubicándola cerca (al lado) de ella.
- **Look Across:** la cardinalidad se representa en el lado opuesto a la entidad, como si estuviéramos leyendo una oración.

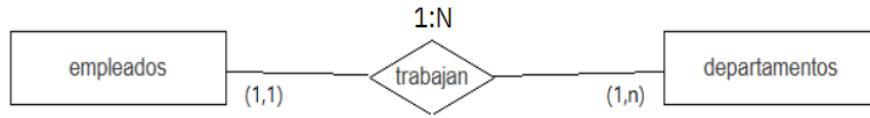
No importa cuál prefieras, siempre y cuando sea coherente en todo el diagrama. Sin embargo, en este tutorial utilizaremos LOOK HERE para el modelo conceptual y el modelo lógico y LOOK ACROSS para el modelo relacional.

Veamos un ejemplo:

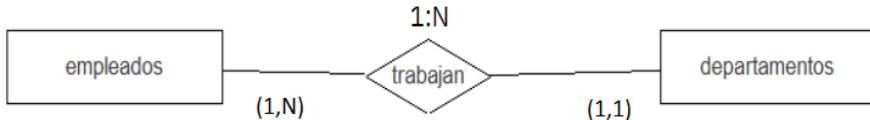
Tenemos la siguiente regla de negocio:

"Un empleado trabaja en un único departamento de forma obligatoria. En el departamento trabajan muchos empleados. Cómo mínimo debe trabajar un empleado en el mismo".

En la Figura 1, se ven las dos notaciones posibles para la cardinalidad en la regla de negocio anterior.



a) Notación Look Here



b) Notación Look Across

Figura 1

Esquema relational

Crear el diagrama

Una vez abierta la aplicación, se ve la ventana que se muestra en la Figura 2.

Aquí se encuentran:

- El ícono para acceder y administrar a las distintas conexiones a servidores de MySQL Server y gestionar el esquema físico de las bases de datos que se encuentren creadas.
- El ícono para acceder y diseñar los distintos esquemas relacionales de bases de datos.
- El ícono para acceder al asistente de migraciones.

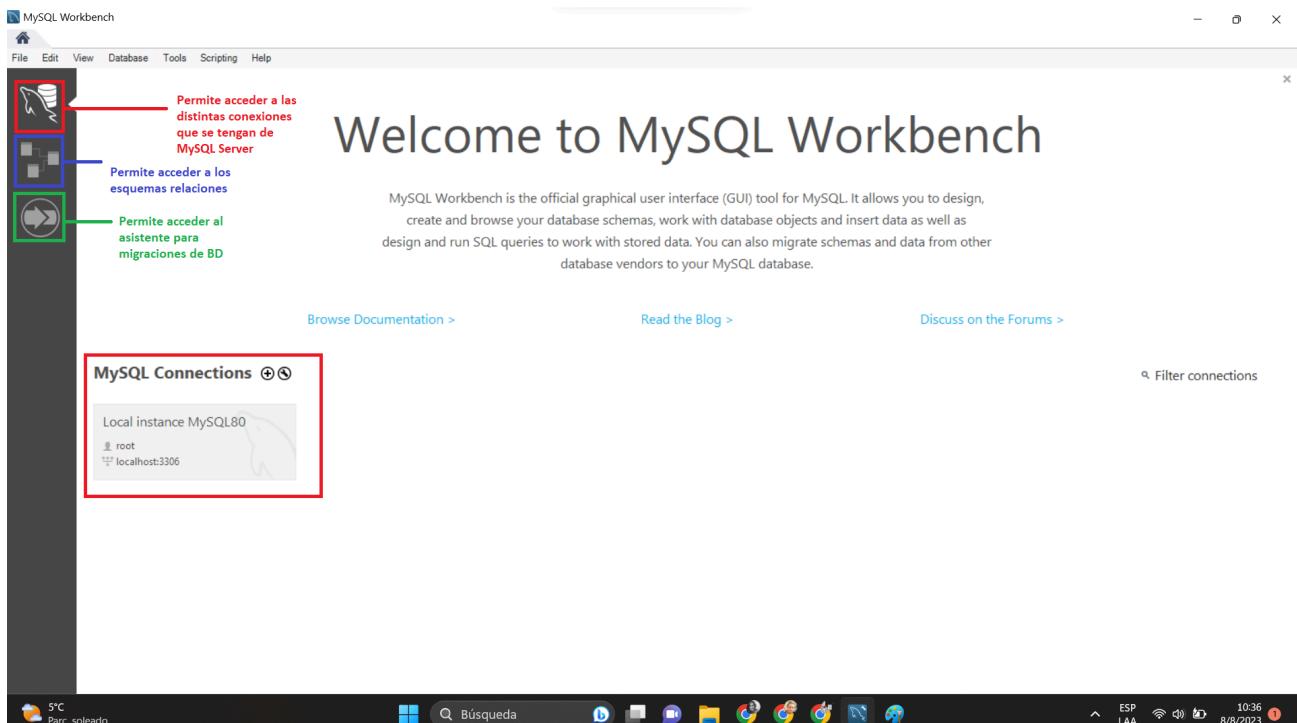


Figura 2

Para crear un nuevo esquema relacional, se hace click sobre el ícono que se muestra en la Figura 3. O bien en el menú File, elegir la opción New Model.

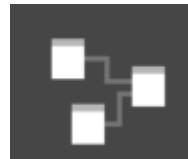


Figura 3

Esto nos conduce a la interfaz que se muestra en la Figura 4:

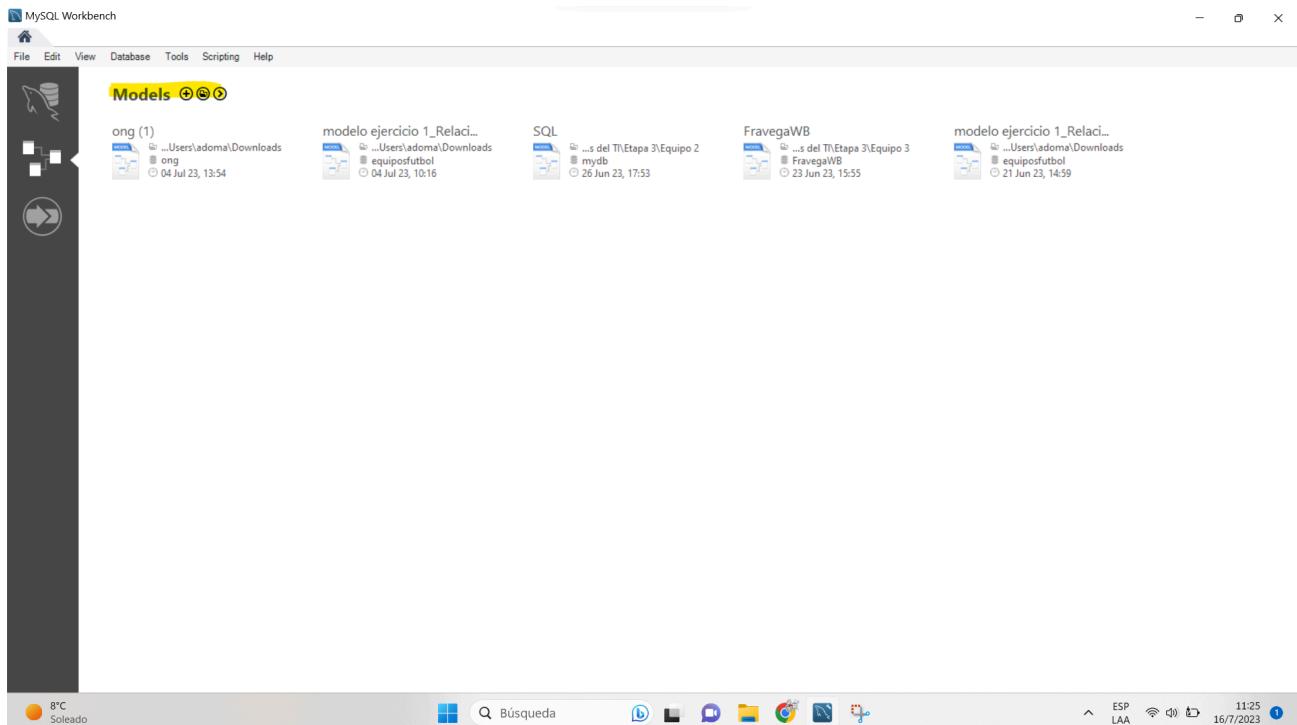


Figura 4

Para crear el diagrama del esquema relacional, se hace click sobre el ícono con el signo +. Esto nos lleva a la interfaz que se muestra en la Figura 5, donde se debe hacer click en el ícono “Add Diagram”.

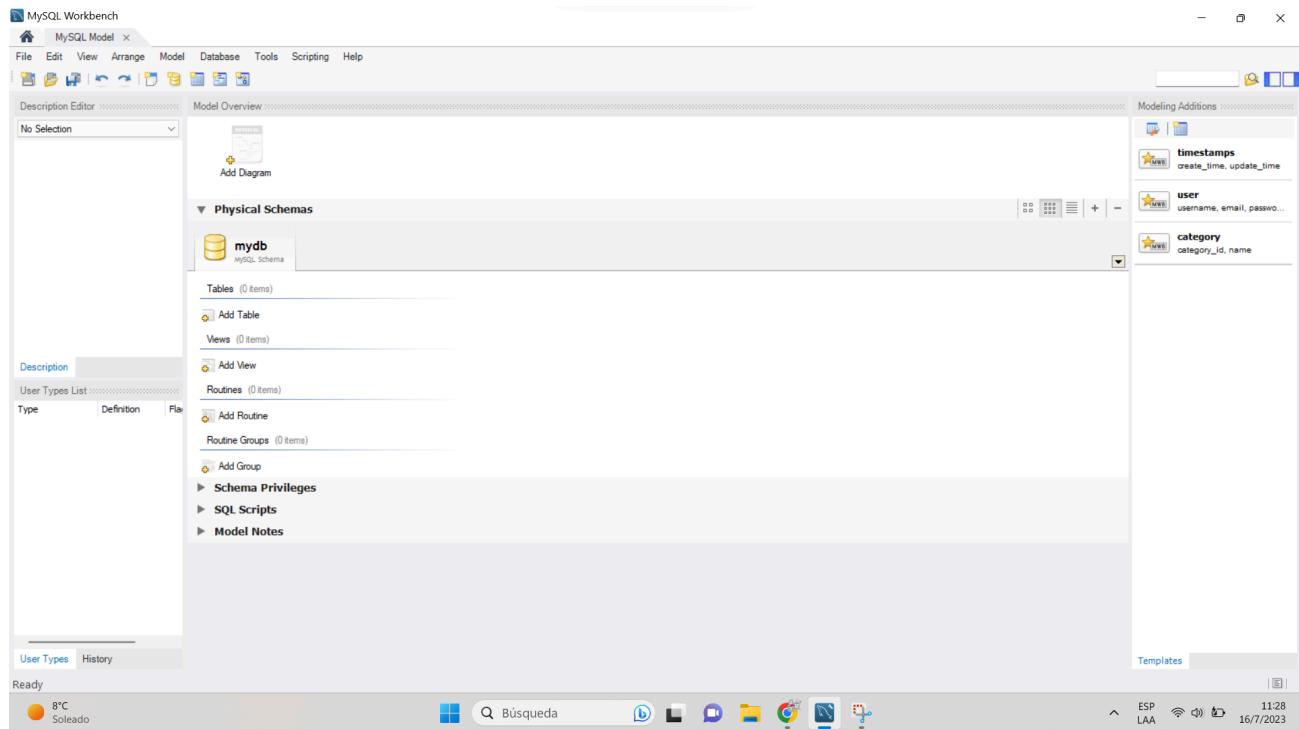


Figura 5

Esta acción nos conduce a la siguiente ventana, donde se pueden comenzar a crear la bases de datos, las tablas del esquema relacional y las relaciones entre ellas. Esto puede verse en la Figura 6.

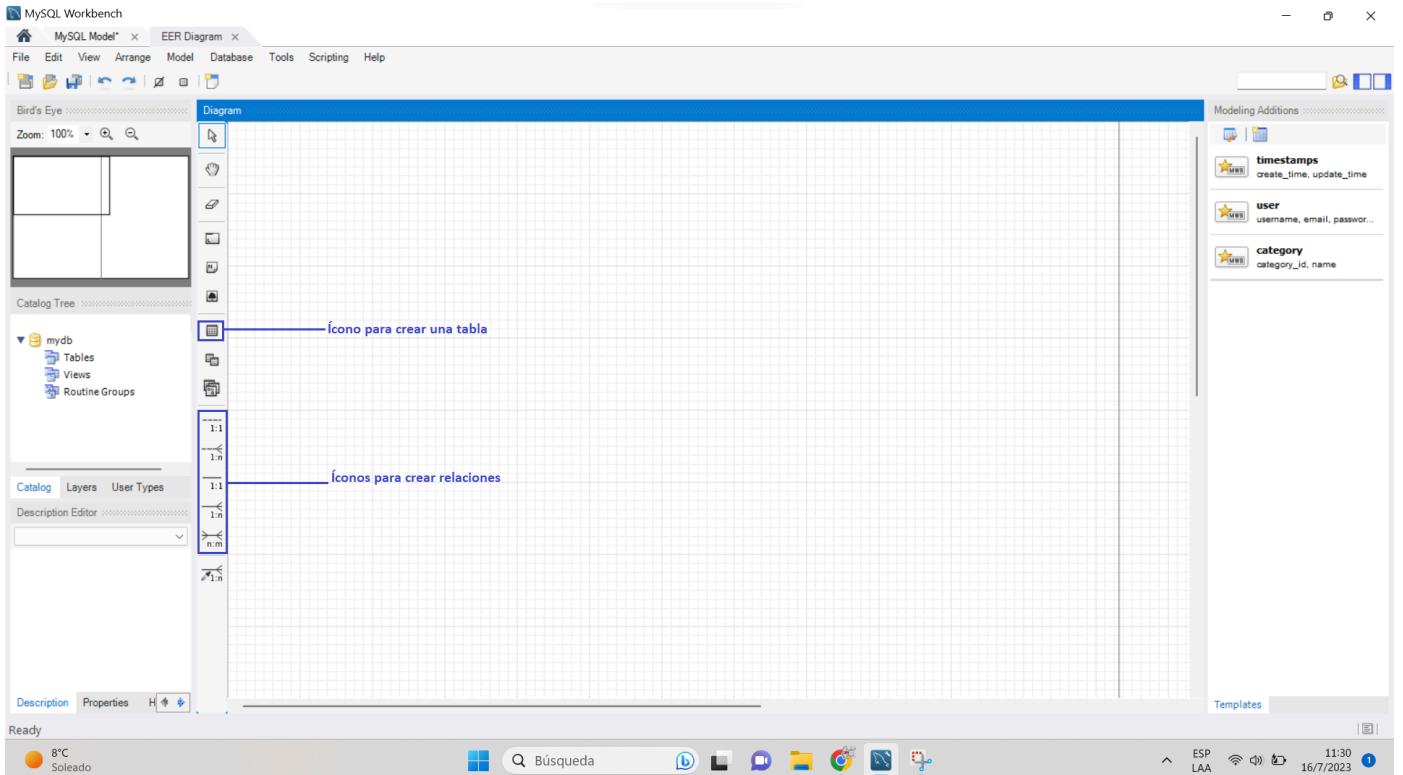


Figura 6

Cambiar el nombre del esquema de la base de datos

Para cambiar el nombre del esquema de la base de datos, debemos posicionarnos sobre el nombre “mydb” como se muestra en la Figura 7 y con el botón derecho del mouse hacer click para editar el esquema. Y luego, completar en el apartado de Name, el nombre de nuestra base de datos.

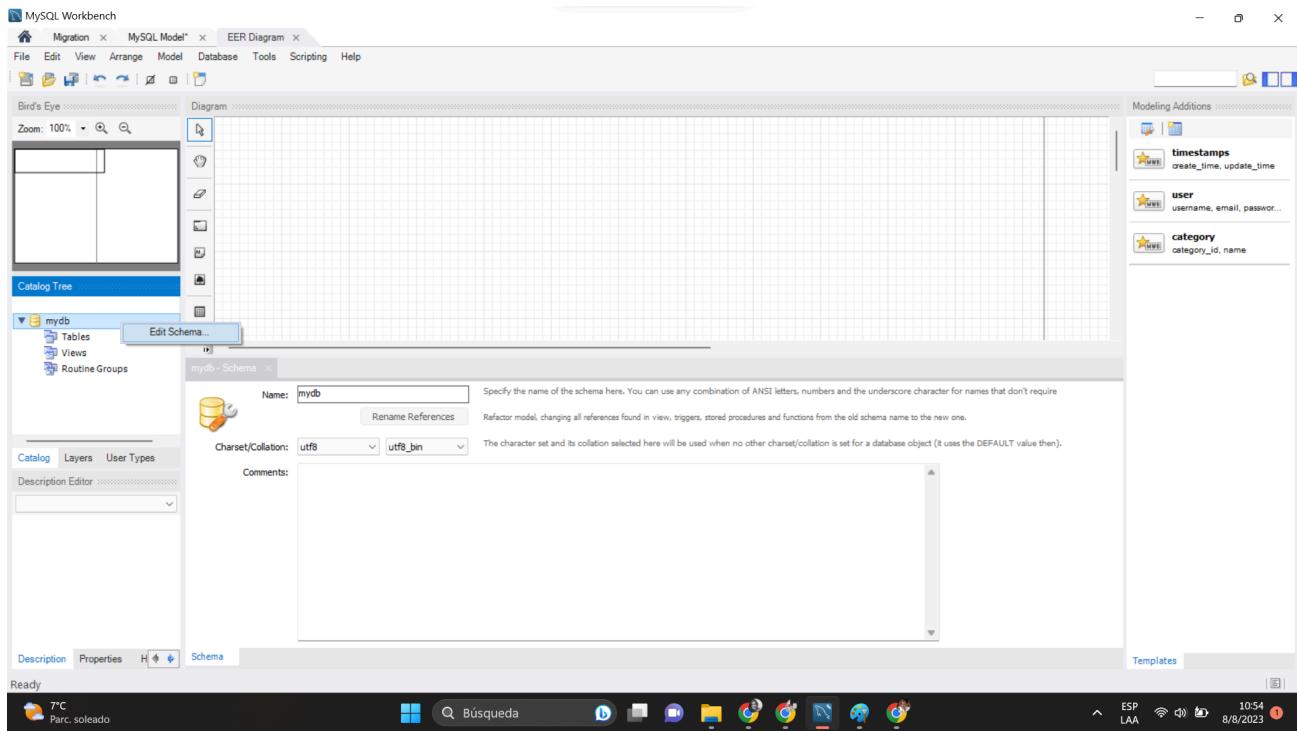


Figura 7

Crear una tabla en el esquema relational

1. Click sobre el ícono Insertar Tabla.
2. Doble Click en la posición de la cuadrícula en la que queremos insertar la tabla. La Figura 8, muestra el resultado de crear una tabla.

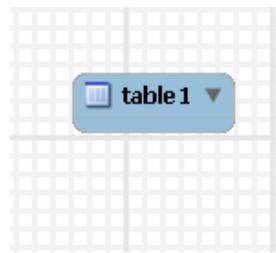




Figura 8

3. Para editar una tabla se puede hacer doble click sobre la tabla o usar el menú de contexto con el botón derecho, opción "Edit Table". En ambos casos se desplegará un menú en la parte inferior como se muestra en las Figuras 9 y 10.

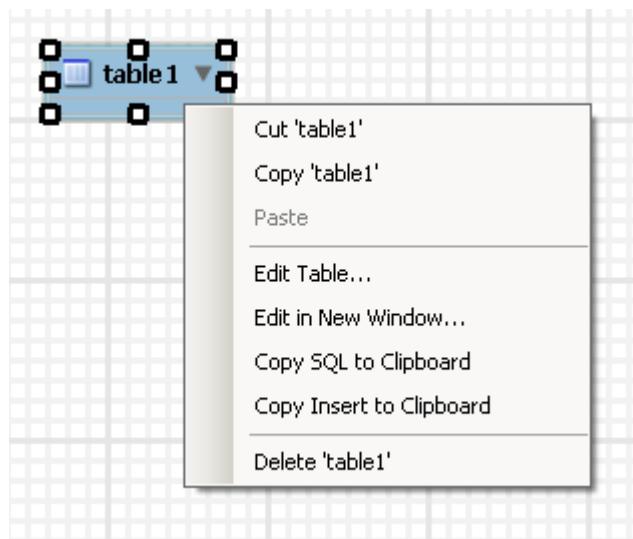


Figura 9



Table Name: table1 Schema: mydb

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
		<input type="checkbox"/>								

Column Name: Data Type: Default: Storage:

Charset/Collation: Comments: Primary Key Not Null Unique Binary Unsigned Zero Fill Auto Increment Generated

Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options | Inserts | Privileges

Figura 10

En la pestaña 'Columns', en la columna 'Table name' se indicará el nombre de la tabla. Por defecto el nombre es 'table1'.

Crear las columnas de una tabla

En la Figura 10, se pueden observar diferentes pestañas que nos permitirán completar las columnas y restricciones de cada tabla.

En la Figura 11 se muestra la pestaña Columns:

Table Name: table1

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idtable1	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
table1col	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 11



Si se hace doble click sobre la primera celda de la columna Column Name, se crea automáticamente la clave primaria entera para la tabla y luego se pueden ir creando el resto de los columnas.

Propiedades de los columnas

Tomemos como ejemplo la tabla países del **Caso de Estudio de Médicos sin Fronteras** (Ver Anexo) de la Figura 12.

países	
!	idpais INT
◆	nombre_pais VARCHAR(45)
◆	grado_conflict INT
◆	continente VARCHAR(100)
Indexes	
	PRIMARY
	nombre_pais_UNIQUE

Figura 12

Cada columna que se crea puede tener las siguientes propiedades:

- **Column Name:** nombre de la columna.
- **Datatype:** dominio de la columna. Se especifica el tipo de dato del mismo.
- **PK:** se utiliza para añadir la restricción de integridad de clave primaria (PK). Se crea automáticamente al crear la tabla, como una clave primaria entera y no nula (NN). Gráficamente, en la tabla, queda marcada con una llave de color amarillo, al lado del nombre del columna. Se crea un índice de PRIMARY KEY automáticamente para la tabla, que puede verse en la columna Indexes.
- **NN:** añade la restricción NOT NULL para esa columna, es decir, que no permitirá valores nulos. Gráficamente, en la tabla, la columna queda marcada con un rombo celeste con relleno, al lado del nombre del mismo. Si no se tilda esta propiedad, el rombo celeste no tiene relleno, indicando que esa columna permite valores nulos.



- **UQ:** añade la restricción UNIQUE para esa columna. Se crea un índice automáticamente para esa columna que puede verse en la pestaña Indexes.
- **AI:** Valor autoincremental. El tipo de dato tiene que ser INT obligatoriamente.
- **Default:** valor por defecto para la columna.

Para añadir una nueva columna, sólo es necesario hacer doble click en la fila que está a continuación de la última fila añadida.

La Figura 13 muestra las propiedades de los columnas para la tabla “paises”.

paises - Table										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idpais	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre_pais	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
grado_conflict	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
continente	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: idpais

Charset/Collation: Default Charset Default Collation

Comments:

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Figura 13

Claves de la tabla

En este apartado veremos los distintos tipos de claves para una tabla en el esquema relacional. Las claves foráneas (FK) las veremos cuando abordemos el tema de las relaciones en el esquema relacional.

Clave Primaria

PK (clave primaria):

- El dominio del atributo toma valores únicos (no se repite). Es UNIQUE.
- Es NOT NULL (no nula).



- Una tabla tiene una única clave primaria, y puede tener varias claves candidatas o no tenerla.

Se crea automáticamente al crear la tabla, como una PK entera y no nula NN. Además, se asume que es única, sin tildar la propiedad UQ para la misma, ya que, se crea un índice de PRIMARY KEY automáticamente para la tabla, que puede verse en la columna Indexes.

Un **índice** es una estructura de datos adicional que permite agilizar el acceso a la información almacenada en un archivo. En dicha estructura se almacenan las claves de los registros del archivo, junto a la referencia de acceso a cada registro asociado a la clave. Es necesario que las claves permanezcan ordenadas¹. En el contexto de las bases de datos, un índice se define sobre una o varias columnas de una tabla, que puede contener claves primarias o secundarias (éstas últimas pueden repetir los valores), de esta forma, se puede acceder de forma más rápida a las filas de la misma. Todos los DBMS relacionales crean índices de forma transparente y automática para los columnas que se definen como PK en el esquema físico. Por eso, MySQL WorkBench lo hace también de forma automática para el esquema relacional.

Gráficamente, en la tabla, queda marcada con una llave de color amarillo, al lado del nombre del columnas (ver Figura 12).

Clave Primaria Subrogada

CS (clave subrogada):

- El dominio del atributo toma valores únicos (no se repite). Es UNIQUE.
- Es NOT NULL (no nula).
- Es Entera (INTEGER).
- Autoincremental (el incremento cada vez que se agrega una tupla o elemento a la tabla es generalmente con paso 1 y lo realiza de forma automática el DBMS).
- No es parte del modelo de entidad relación (modelado conceptual). Esta clave no es semántica, es decir, no tiene significado en el universo del discurso.
- Se agrega al modelo Relacional para elegirla como PK.

¹ Bertone, R.; Thomas, P. (2011) Introducción a las Bases de Datos. Fundamentos y Diseño. Primera Edición, Buenos Aires. Prentice Hall - Pearson Education.



Para usar claves primarias subrogadas, hay que tildar además de lo que se crea por defecto para la columna que es PK, la propiedad para que ésta sea autoincrementable (AI), esto la convierte en una **PK subrogada (entera, no nula, única y autoincremental)**. En la Figura 14, se observa la clave primaria subrogada **idpais** para la tabla “paises” del **Caso de Estudio de Médicos sin Fronteras**.

PAISES - Table										
	Table Name:	PAISES								
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idpais	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
nombre_pais	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
grado_conflict	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
continente	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 14

Claves primarias compuestas

Cuando no se usan las PK subrogadas para las tablas, se puede usar esta opción, que se corresponde con la teoría formal del modelo relacional.

Cabe aclarar que esta opción será válida sólo para aquellas tablas que tengan claves compuestas en las entidades correspondientes en el esquema lógico para las tablas resultantes de la transformación al esquema relacional de las relaciones N:M o relaciones ternarias.

En este tutorial siempre usaremos **PK subrogadas** por una cuestión de eficiencia.

Los pasos a seguir para crear una **PK compuesta** son los siguientes:

1. Hacer doble click sobre la tabla.
2. Abrir la pestaña 'Columns'.
3. Seleccionar la columna que se desea utilizar como clave primaria.
4. Marcar la opción PRIMARY KEY.
5. Si se desea definir una clave primaria compuesta, sobre más de una columna, repetir las acciones desde el punto 3.
6. No deben ser incrementales (AI).



Claves Candidatas (o secundarias)

CK (clave candidata):

- Es un atributo identificador del modelo de entidad relación (Esquema Lógico).
- El dominio del atributo toma valores únicos (no se repite). Es UNIQUE.
- Es NOT NULL (no nula).
- Es candidata a ser elegida como PRIMARIA.

Para mantener la trazabilidad entre el esquema lógico y el esquema relacional, todos los atributos identificadores de las entidades en el esquema lógico, deberán ser claves candidatas (CK), es decir, columnas únicas (UQ) y no nulas (NN) en el esquema relacional.

Claves candidatas simples

En la Figura 15, se observa la columna **nombre** que es una CK simple para la tabla "paises". La misma tiene marcadas las propiedades de unicidad (UQ) y no permite valores nulos (NN).

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idpais	INT									
nombre	VARCHAR(100)		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						
grado_conflict	INT									
continente	VARCHAR(100)									

a) Restricción de clave candidata para la columna nombre de la tabla paises

Index Name	Type	Index Columns																
PRIMARY	PRIMARY																	
ck_paises_nombre	UNIQUE	<table border="1"> <thead> <tr> <th>Column</th> <th>#</th> <th>Order</th> <th>Length</th> </tr> </thead> <tbody> <tr> <td>nombre</td> <td>1</td> <td>ASC</td> <td></td> </tr> <tr> <td>grado_conflict</td> <td></td> <td>ASC</td> <td></td> </tr> <tr> <td>continente</td> <td></td> <td>ASC</td> <td></td> </tr> </tbody> </table>	Column	#	Order	Length	nombre	1	ASC		grado_conflict		ASC		continente		ASC	
Column	#	Order	Length															
nombre	1	ASC																
grado_conflict		ASC																
continente		ASC																

b) Usar un nombre representativo para la restricción de CK (buena práctica).

Figura 15



Claves candidatas compuestas

En la Figura 16 se observa la clave candidata compuesta para la tabla “lugares” del **Caso de Estudio de Médicos sin Fronteras**

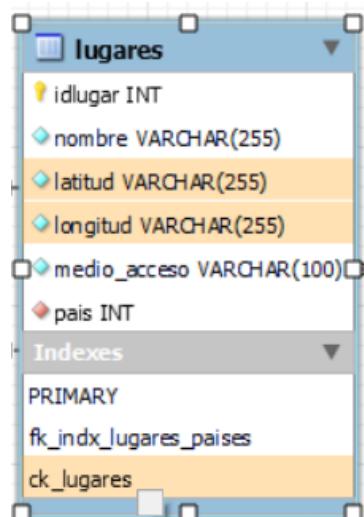


Figura 16

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idlugar	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
latitud	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
longitud	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 17

Como puede verse en la Figura 17, ambas columnas (**latitud** y **longitud**) no permiten valores nulos (NN), pero no son únicas (UQ). Para la condición de unicidad de la CK



compuesta debe crearse un índice de tipo UNIQUE para ambas columnas conjuntamente, como se ve en la Figura 18. Esto significa que cada combinación de valores para latitud y longitud será única.

Index Name	Type
PRIMARY	PRIMARY
fk_idx_lugares_paises	INDEX
nombre_UNIQUE	UNIQUE
ck_lugares	UNIQUE

Index Columns			
Column	#	Order	Length
latitud	2	ASC	
longitud	1	ASC	

Figura 18

Columnas de la tabla y tipos de datos

Las columnas de las tablas que no son CK, es decir, que eran atributos descriptores en el esquema lógico, se irán completando según las propiedades que tengan y los tipos de datos.

Para elegir el tipo de dato se deben tener en cuenta los tipos implementados por el Sistema Gestor de Bases de Datos (SGBD) MySQL Server.

Aprende más sobre tipos de datos en MySQL en el siguiente link:

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

En la tabla se verán gráficamente con rombos celestes con relleno y sin relleno, dependiendo de la propiedad NOT NULL (NN).

En la Figura 19, continuando con la tabla “paises”, se observan los columnas **grado_conflict** y **continente** para la tabla “paises”. El columna **grado_conflict** es de tipo entero (INT), permite nulos, puede solo tomar valores enteros entre 0 y 10 y siempre por



defecto es cero. Es decir, tiene un dominio definido por el usuario (números enteros entre 0 y 10) y, además, un valor por defecto. Esta restricción de dominio a nivel de columna (CHECK) la veremos cuando abordemos el lenguaje SQL DDL, no puede marcarse en esta instancia del diseño en MySQL WorkBench. La columna **continente** es un varchar con longitud de 100 caracteres y no permite nulos.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idpais	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
nombre_pais	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
grado_conflict	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
continente	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 19

Las columnas que son *calculadas* deben tener siempre el valor por defecto en cero, como por ejemplo, la **cantidad_total** en la tabla “voluntarios” es de tipo NUMERIC con 4 dígitos antes de la coma y dos después de la coma. Además, al ser calculada, tiene por defecto el valor cero (Figura 20).

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
apellido_vol	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
cantidad_total	NUMERIC(4,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0					
pais	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figura 20



Dominios definidos por el usuario

En MySQL, no se admiten los comandos `CREATE TYPE` o `CREATE DOMAIN` para definir tipos de datos personalizados, como ocurre en otros sistemas de gestión de bases de datos (SGBD) que cumplen con el estándar ANSI SQL más avanzado. Sin embargo, se pueden usar los tipos de datos propios del SGBD como `SET` o `ENUM`, pero éstos no son parte del estándar ANSI SQL.

A instancias de este tutorial, para el esquema relacional en MySQL WorkBench no utilizaremos dominios definidos por el usuario.

Relaciones entre las tablas

La notación que se usa para las relaciones en este tutorial, es la que se encuentra elegida por defecto Crow's Foot (IE). Para cambiarla, se debe ir al menú superior, elegir la opción Model, luego la opción Relationship Notation y tildar la opción que se desee usar.

Tipos de relaciones

En la Figura 21, se muestran los diferentes tipos de relaciones.



Figura 21



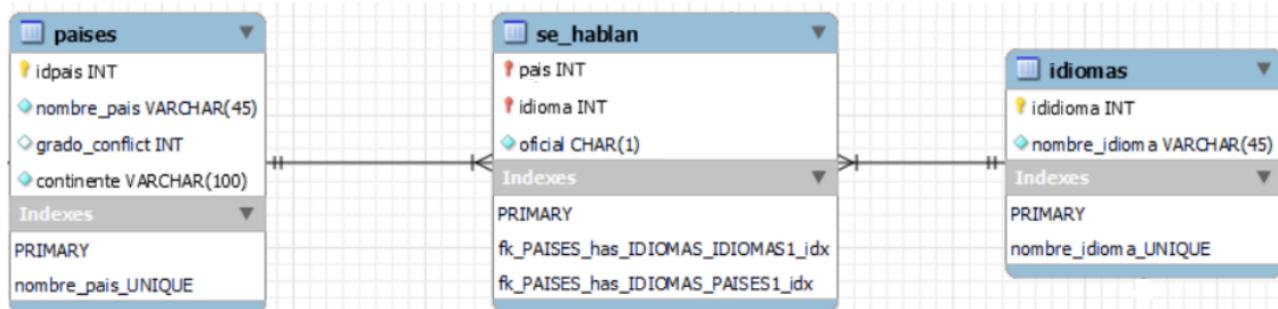
- **Relación 1:1 No identificatoria:** propaga la PK de una tabla como FK a otra tabla. Se usa para relaciones con conectividad 1:1.
- **Relación 1:N No identificatoria:** propaga la PK de una tabla como FK a otra tabla. Se usa para relaciones con conectividad 1:N.
- **Relación 1:1 Identificatoria:** propaga la PK de una tabla como FK a otra tabla, pero además será PK no subrogada. Se usa para relaciones con conectividad 1:1.
- **Relación 1:N Identificatoria:** propaga la PK de una tabla como FK a otra tabla, pero además será PK no subrogada. Se usa para relaciones con conectividad 1:N.
- **Relación N:M:** crea una nueva tabla para las relaciones con conectividad N:M.

Relaciones con conectividad N:M

Para crear una relación con conectividad N:M, MySQL WorkBench lo implementa como se indica en la teoría del modelo relacional, para lo cual se deben seguir los siguientes pasos:

1. Hacer click sobre el ícono de la relación N:M.
2. Hacer click sobre una de las tablas que componen la relación (no importa en el orden, pues es una relación N:M)
3. Hacer click sobre la otra tabla.
4. Se crea una nueva tabla como se muestra en la Figura 22.

En el **Caso de Estudio de Médicos sin Fronteras**, tenemos la regla de negocio: “En los países se hablan distintos idiomas y cada idioma se habla al menos en un país. Cada país tiene como mínimo un idioma oficial, se quieren saber tanto los idiomas oficiales como los no oficiales”.





- a) Nueva tabla por la relación N:M “se_hablan” entre las tablas países e idiomas con PK compuesta

se_hablan - Table							
Column Name	Datatype	PK	NN	UQ	B	UN	ZF
idse_hablan	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
pais	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
idioma	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
oficial	CHAR(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name: Data Type: Default:

Charset/Collation:

Comments:

Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

- b) Detalle de las propiedades de las columnas de la tabla “se_hablan”

Figura 22

En esta regla de negocio vemos que la PK compuesta de la tabla nueva “se_hablan”, está formada por las PK de las dos tablas involucradas en la relación (pais + idiomas), son enteras, no nulas y *no son autoincrementales*. Además, la tabla contiene el atributo **oficial** que toma los valores del conjunto {S, N}, dependiendo de si se trata de su idioma oficial o no.

Claves Foráneas

Cada una de los columnas **pais** e **idiomas** son claves foráneas (FK) de manera individual para la tabla “se_hablan”.

Una FK (clave foránea o ajena):

- Es un atributo en la tabla que hace referencia a la clave primaria (PK) de otra tabla.
- Una FK puede ser nula.
- No necesariamente es única. Generalmente sus valores se repiten.

En la pestaña Foreigns Key para las propiedades de la tabla se_hablan (Figura 23), se observan los nombres de cada FK de la tabla, como la columna de la misma que tiene esa restricción.



se_hablan - Table

Table Name: se_hablan Schema: ong

Foreign Key Name Referenced Table
fk_paises_se_hablan 'ong'. 'paises'
fk_idiomas_se_hablan 'ong'. 'idiomas'

Column	Referenced Column
<input type="checkbox"/> idse_hablan	
<input checked="" type="checkbox"/> pais	idpais
<input type="checkbox"/> idioma	
<input type="checkbox"/> oficial	

Foreign Key Options
On Update: NO ACTION
On Delete: NO ACTION
 Skip in SQL generation
Foreign Key Comment

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

c) Restricción de clave foránea para la columna pais

se_hablan - Table

Table Name: se_hablan Schema: ong

Foreign Key Name Referenced Table
fk_paises_se_hablan 'ong'. 'paises'
fk_idiomas_se_hablan 'ong'. 'idiomas'

Column	Referenced Column
<input type="checkbox"/> idse_hablan	
<input type="checkbox"/> pais	
<input checked="" type="checkbox"/> idioma	idioma
<input type="checkbox"/> oficial	

Foreign Key Options
On Update: NO ACTION
On Delete: NO ACTION
 Skip in SQL generation
Foreign Key Comment

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

d) Restricción de clave foránea para la columna idioma

Figura 23

Foreign Key (FK) Name: nombre de la restricción de clave foránea. Gráficamente se marca con un rombo rojo relleno, ya que, por defecto las claves foráneas son NOT NULL (lo que no siempre es aplicable). Si las claves foráneas, además son claves primarias de la tabla, como en este caso, se marcan con una llave roja (Figura 22).

Siempre es una buena práctica renombrar las columnas que tienen una restricción de FK con los nombres que poseían en el esquema lógico (Figura 22).

Siempre es una buena práctica renombrar las restricciones de FK con nombres que sean más representativos y fáciles de recordar (Figura 23).

En la Figura 23, se muestran las restricciones de FK renombradas y las columnas de la tabla se_hablan en los cuales se aplica esa restricción. También se observan las acciones a llevar a cabo para las opciones On Update y On Delete.

Recordemos que los valores para las distintas acciones que pueden tomar las opciones On Update y On Delete son: NO ACTION, RESTRICT, SET DEFAULT, SET NULL y CASCADE.



MySQL permite solo las acciones que se observan en la Figura 24.

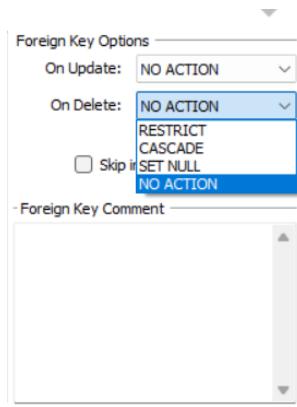


Figura 24

Índices automáticos para FK

Como puede verse en las Figura 25 y Figura 26, MySQL WorkBench crea de forma automática un índice asociado a la columna de la tabla a la cual se le define una restricción de clave foránea (FK). No todos los DBMS se comportan de la misma forma. Se deberá chequear para cada caso según corresponda.

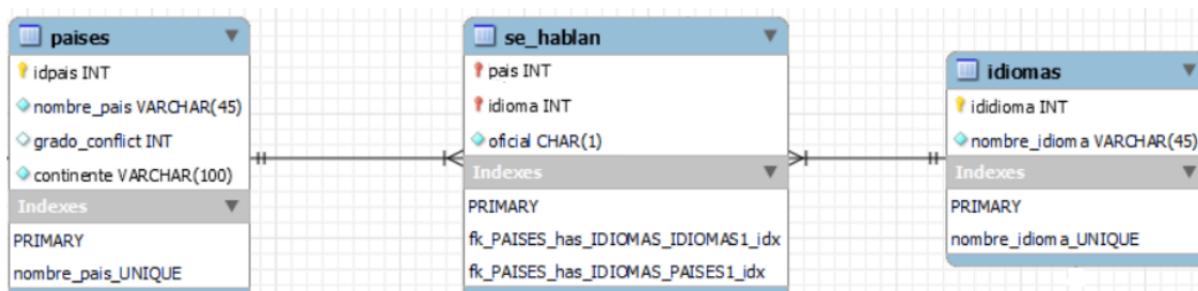


Figura 25



se_hablan - Table

Table Name: se_hablan Schema: ong

Index Name	Type
PRIMARY	PRIMARY
fk_idx_idiomas_se_hablan	INDEX
fk_idx_paises_se_hablan	INDEX
ck_se_hablan	UNIQUE

Index Columns

Column	#	Order	Length
<input type="checkbox"/> idse_hablan		ASC	
<input type="checkbox"/> pais		ASC	
<input checked="" type="checkbox"/> idioma	1	ASC	
<input type="checkbox"/> oficial		ASC	

Index Options

- Storage Type:
- Key Block Size: 0
- Parser:
- Visible:

Index Comment

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

a) Índice para la clave foránea de la columna idioma

se_hablan - Table

Table Name: se_hablan Schema: ong

Index Name	Type
PRIMARY	PRIMARY
fk_idx_idiomas_se_hablan	INDEX
fk_idx_paises_se_hablan	INDEX
ck_se_hablan	UNIQUE

Index Columns

Column	#	Order	Length
<input type="checkbox"/> idse_hablan		ASC	
<input checked="" type="checkbox"/> pais	1	ASC	
<input type="checkbox"/> idioma		ASC	
<input type="checkbox"/> oficial		ASC	

Index Options

- Storage Type:
- Key Block Size: 0
- Parser:
- Visible:

Index Comment

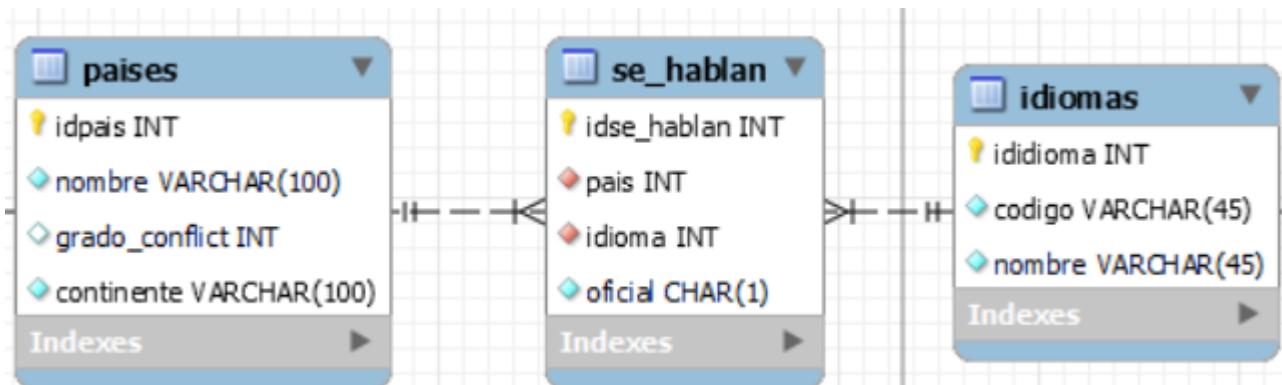
Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

b) Índice para la clave foránea de la columna pais
Figura 26

Usando PK subrogada para las relaciones N:M

Otra manera de resolver las relaciones N:M es agregarle a la nueva tabla una clave primaria (PK) subrogada (CS), con lo cual, para mantener la trazabilidad entre el esquema lógico y el relacional, la PK compuesta que se crea por definición del modelo relacional, ahora pasará a ser clave candidata compuesta (CK), es decir, única y no nula.

En la figura 27, puede verse la nueva tabla por la relación N:M “se_hablan” entre las tablas “paises” e idiomas del **Caso de Estudio de Médicos sin Fronteras**, con la PK subrogada **idse_hablan** y las columnas **idpais** e **idiomas** cada una como columnas no nulas, tener restricción de clave foránea y conjuntamente como conformar una clave candidata compuesta con un índice de tipo UNIQUE.



- a) Nueva tabla por la relación N:M “se_hablan” entre las tablas países e idiomas con PK subrogada y CK compuesta

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idse_hablan	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
pais	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
idioma	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
oficial	CHAR(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					

- b) Propiedades de las columnas pais e idioma de la tabla “se_hablan”, que conforman la CK compuesta y son FK

- c) Índice de tipo UNIQUE “ck_se_hablan” para la tabla “se_hablan” que permite crear la CK compuesta para las columnas pais e idioma

Figura 27



Relaciones con conectividad 1:N

Propagación de clave

Cardinalidad (1,1) → (1, N) o (1,1) → (0, N)

Si en la relación binaria 1:N, la entidad que participa con cardinalidad máxima uno lo hace con cardinalidad mínima uno entonces:

1. Hacer click sobre el ícono de la relación 1:N No Identificatoria.
2. La manera en que se resuelve la relación 1:N es propagando la clave de la tabla que participa con cardinalidad (1,N) a la tabla que participa con cardinalidad (1,1), así como los atributos de la relación, si los hubiere. La clave propagada será una clave foránea (FK).
3. Hacer click sobre la tabla que participa con cardinalidad (1,1).
4. Hacer click sobre la otra tabla, la que participa con cardinalidad (1,N).

En la Figura 28, podemos ver la que se define la relación con conectividad 1:N “está_ubicado” entre países y lugares del **Caso de Estudio de Médicos sin Fronteras**, sabiendo que un lugar está ubicado en un solo país y en un país se encuentran muchos lugares.

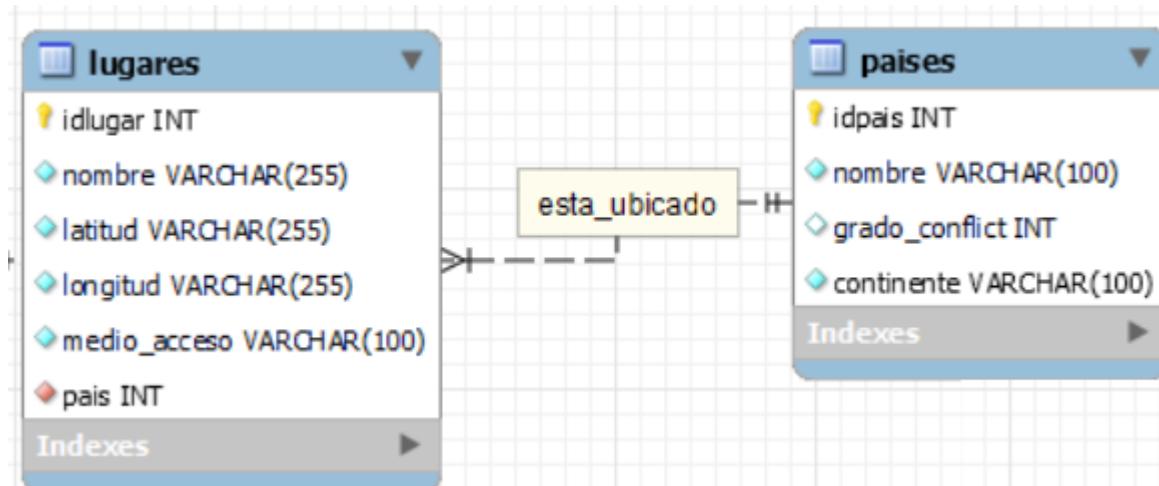
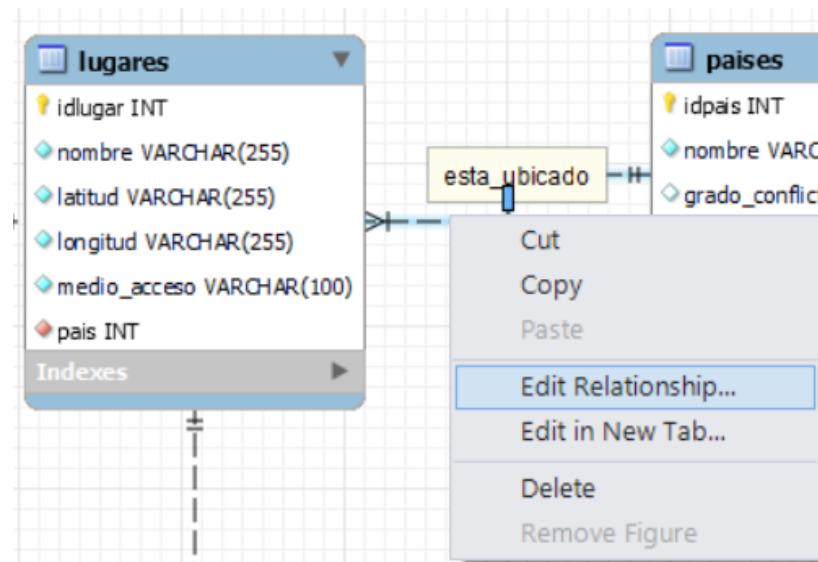


Figura 28



A la relación resultante de la resolución de la relación 1:N, se le puede cambiar el nombre (Caption), con botón derecho, menú de contexto, “Edit Relationship”, para que tenga el mismo nombre que la relación del esquema Lógico (Figura 29), como también se puede cambiar el nombre de la FK de la tabla “lugares”, como se vio en el apartado sobre este tema.



a) Menú de contexto para editar la relación



b) Cambiando el caption de la relación
Figura 29



Cardinalidad (0,1) → (1, N) o (0,1) → (0, N)

Si en la relación binaria 1:N, la entidad que participa con cardinalidad máxima uno lo hace con cardinalidad mínima cero entonces:

Se resuelve como en el punto anterior, pero la columna que contiene la restricción de clave foránea (FK) -clave propagada- deberá permitir valores NULOS.

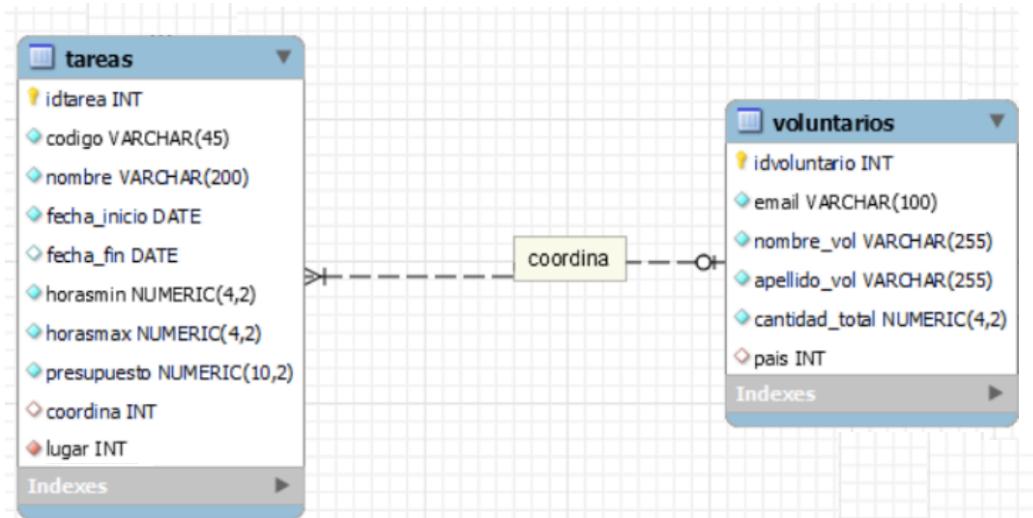


Figura 30

En la Figura 30, continuando con el **Caso de Estudio de Médicos sin Fronteras**, vemos que las tareas pueden tener o no un coordinador y éste es un voluntario. Además, un voluntario puede coordinar ninguna o muchas tareas.

Figura 31



En la tabla “tareas” la columna coordinador que tiene la restricción de clave foránea debe permitir valores nulos. Aparecerá con el rombo rojo sin llenar y no deberá tildarse la bandera NN. Esto puede verse en la Figura 31.

Creando una nueva tabla

Cardinalidad $(0,1) \rightarrow (1, N)$ o $(0,1) \rightarrow (0, N)$

Otra forma de resolver las relaciones con conectividad 1:N cuando la entidad que participa con cardinalidad máxima uno lo hace con cardinalidad mínima cero, es crear una nueva tabla para la relación, que tendrá:

- una columna de tipo INT que tendrá una restricción de PK (no subrogada) para la nueva tabla y una restricción de FK que referencia a la PK de la tabla que participa con cardinalidad máxima 1.
- una columna de tipo INT que tendrá una restricción de FK que referencia a la PK de la tabla que participa con cardinalidad máxima N.
- una columna para cada uno de los atributos de la relación si los hubiera, teniendo en cuenta las restricciones de los mismos en el esquema lógico.

En la Figura 32 puede verse la nueva tabla creada para la relación “coordina” entre las tablas “tareas” y “voluntarios”.

Para resolver esto debemos crear la nueva tabla para la relación “coordina” y utilizar para la columna “tarea” la relación 1:N identificable, es decir la FK propagada desde la tabla “tareas” ES TAMBIÉN PK de la nueva tabla y la relación 1:N no identificable para la columna “voluntario”, es decir, la FK propagada desde la tabla “voluntarios” NO es PK de la nueva tabla.

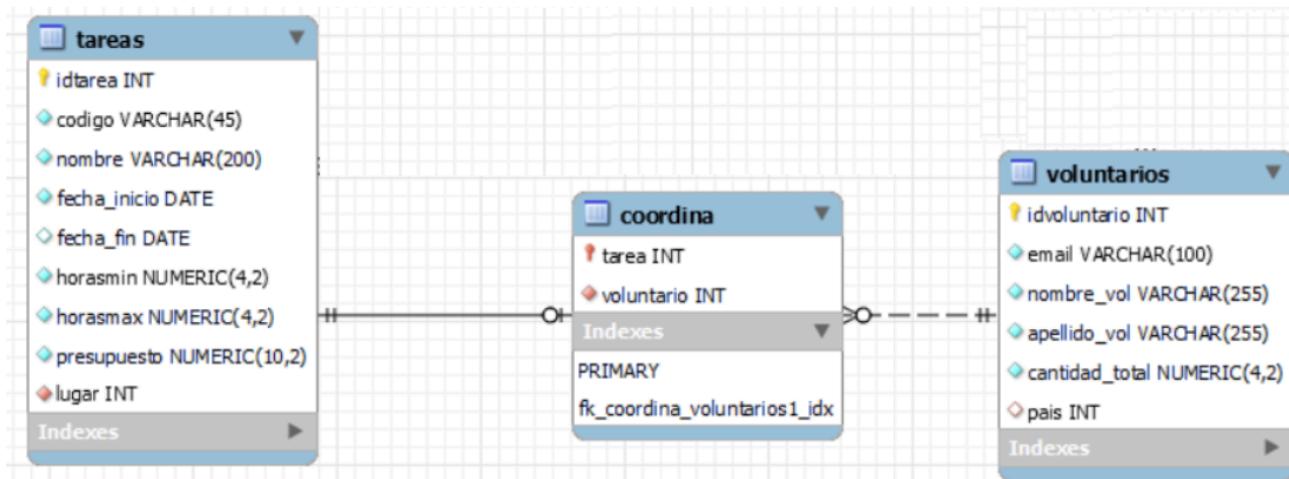


Figura 32

Relaciones con conectividad 1:1

Las relaciones con conectividad 1:1 son un caso especial de las relaciones con conectividad 1:N, donde N=1, por ende, se deben utilizar los casos de resolución de relaciones con conectividad 1:N, analizando qué caso aplicar según las cardinalidades mínimas.

Relaciones Mandatorias y No Mandatorias.

¿Cómo marcar la cardinalidad mínima cero de las relaciones?

Una relación según su cardinalidad mínima puede ser **mandatoria** (con participación obligatoria) o **no mandatoria** (con participación optativa).

Las relaciones mandatorias, son aquellas cuya cardinalidad mínima es uno. Es obligatoria la participación de las entidades en la relación.

Las relaciones no mandatorias, son aquellas cuya cardinalidad mínima es cero. No es obligatoria la participación de las entidades en la relación.

Para definir la cardinalidad en WorkBench, de forma que exista trazabilidad entre el esquema lógico y el esquema relacional, debemos pararnos con el mouse sobre la relación a la cual queremos modificar la cardinalidad, con el botón derecho del mouse, abrir el menú de contexto y elegir la opción “Edit Relationship”, como se muestra en la Figura 33, luego se abre la ventana que se muestra en la Figura 34.

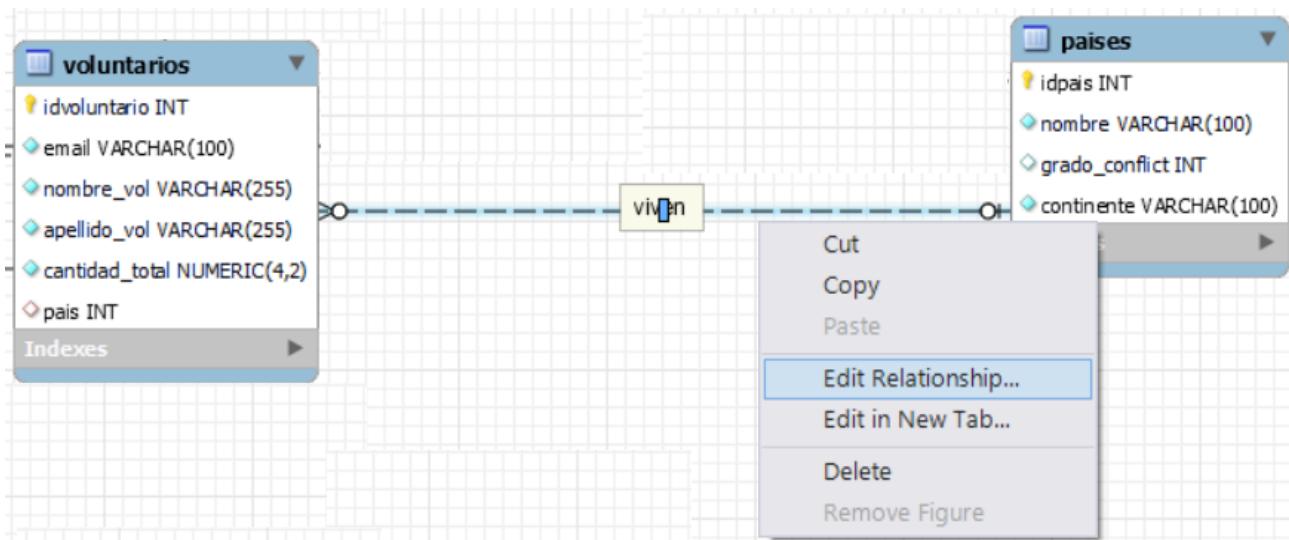


Figura 33

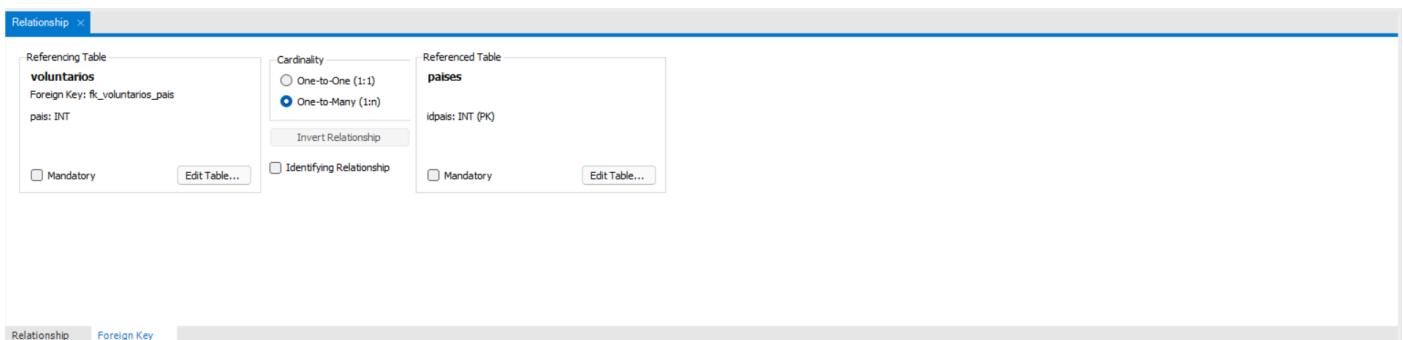


Figura 34



En la pestaña de Foreign Key, se debe seleccionar o tildar (mandatoria) o destildar (no mandatoria) la opción Mandatory para la tabla correspondiente. Recordar que aquí se usa la notación look across, por eso, debe leerse de forma cruzada.

Esto debe hacerse para todas las relaciones del esquema relacional, en función de las cardinalidades del esquema lógico.

En el ejemplo de la Figura 34 del **Caso de Estudio de Médicos sin Fronteras**, la relación “viven” entre voluntarios y países es no obligatoria para ambas tablas, ya que no necesariamente todos los países tendrán un voluntario que sea originario de él, y además, se tiene la restricción semántica de que el país de origen del voluntario puede no saberse. Además, de un país pueden ser originarios muchos voluntarios, pero un voluntario es originario como máximo de un solo país. Con lo cual, es una *relación con conectividad 1:N no mandatoria* para ambas tablas.

Etiquetar las relaciones

Para utilizar las etiquetas, debemos utilizar el ícono que se muestra en la Figura 35.



Figura 35

Esta acción nos permite crear una etiqueta y ubicarla en cualquier lugar del diagrama. En la Figura 36 se muestra el formulario para la creación de la etiqueta de la relación “viven” del apartado anterior.

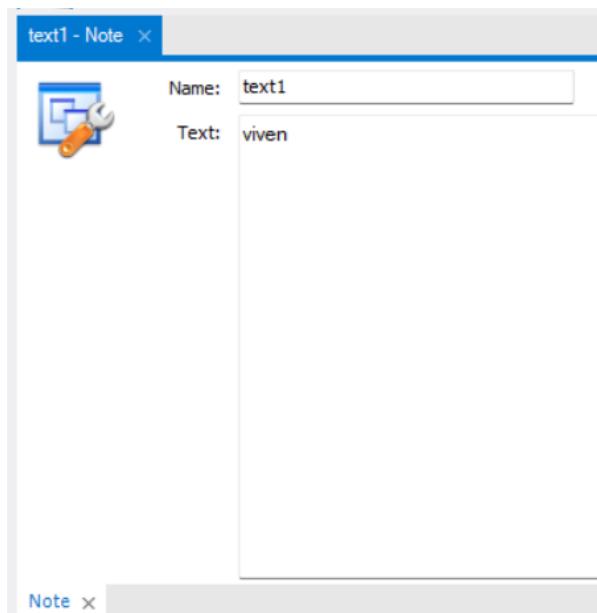


Figura 36

Jerarquías (“Es_un”) - Relaciones 1:1 Identificables

En el esquema lógico las jerarquías se pueden resolver con alguna de las siguientes tres opciones:

1. Eliminar a la superentidad
2. Eliminar a las subentidades
3. Dejar todo y hacer explícitas las relaciones “es_un” dadas por la herencia.

La resolución de los casos 1 y 2, no conlleva demasiada complejidad, ya que en esquema relacional se tratarán a las entidades como cualquier otra tabla. Sin embargo, cuando utilizamos el caso 3, las relaciones “es_un” solo deben resolverse usando el tipo de **relación 1:1 Identifiable**, es decir, la FK propagada a las subentidades desde la superentidad será además PK de las mismas y será no autoincrementable. La PK de la superentidad si será una PK subrogada.

En la Figura 37, puede verse la resolución de la jerarquía del **Caso de Estudio de Médicos sin Fronteras**, donde un voluntario puede ser civil o médico.

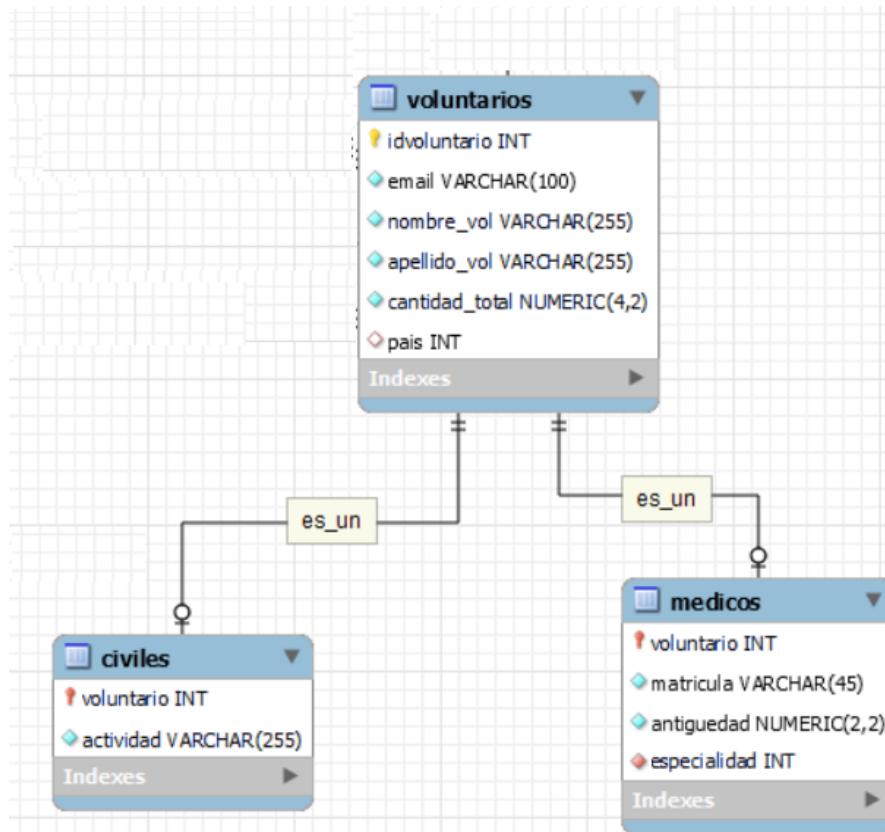


Figura 37

En la Figura 38, se observa que en la superentidad “voluntarios” tiene como PK subrogada a la columna “idvoluntario”.



voluntarios - Table

Table Name: voluntarios Schema: ong

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idvoluntario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
email	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre_vol	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
apellido_vol	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: idvoluntario Data Type: INT Default:

Charset/Collation: Default Charset Default Collation

Comments:

Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Figura 38

En la Figura 39, se observa que la subentidad “civiles” tiene la columna “voluntario” que es al mismo tiempo FK y PK no subrogada.

civiles - Table

Table Name: civiles Schema: ong

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
voluntario	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
actividad	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
civilescol	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: voluntario Data Type: INT Default:

Charset/Collation: Default Charset Default Collation

Comments:

Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Figura 39

En la Figura 40, se observa que la subentidad “medicos” tiene la columna “voluntario” que es al mismo tiempo FK y PK no subrogada.



medicos - Table

Table Name: medicos Schema: ong

Column Name Datatype PK NN UQ B UN ZF AF G Default/Expression

voluntario INT

matricula VARCHAR(45)

antiguedad NUMERIC(2,2)

especialidad INT

Column Name: voluntario Data Type: INT
Charset/Collation: Default Charset Default Collation
Comments:

Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

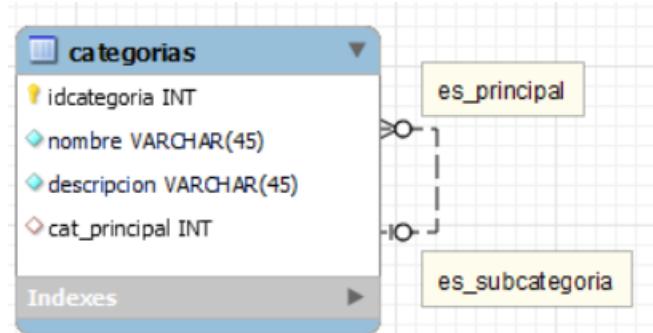
Figura 40

Relaciones Recursivas

Las relaciones recursivas son una relación binaria de una entidad consigo misma. La conectividad de la relación dependerá de la regla de negocio que da origen a la misma.

Si tenemos la regla de negocio para las categorías de los productos de una tienda: “Una categoría tiene un nombre único y una descripción. Además, puede incluir subcategorías. Una subcategoría se encuadra en una sola categoría principal”.

En la Figura 41, vemos la relación recursiva “subcategoria”, donde una categoría puede contener a otras categorías, es decir, es una categoría principal o una categoría puede ser subcategoría de otra categoría. En este caso, se resuelve utilizando la opción de propagación de clave, ya que la conectividad es 1:N.





a) Tabla categorías

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idcategoria	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
description	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
cat_principal	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: nombre
Charset/Collation: Default Charset
Comments:

Data Type: VARCHAR(45)
Default:
Storage: Virtual
 Primary Key
 Not Null
 Unique
 Binary
 Unsigned
 Zero Fill
 Auto Increment
 Generated

b) Columnas de la tabla categorías

Referencing Table: categorias
Foreign Key: FK_IdSubcategorias
cat_principal: INT

Cardinality: One-to-One (1:1) (radio button)
One-to-Many (1:n) (radio button selected)

Invert Relationship
 Identifying Relationship

Referenced Table: categorias
idcategoria: INT (PK)

Mandatory
 Edit Table...

c) Cardinalidad de la tabla categorías

Figura 41

En el ejemplo del **Caso de Estudio de Médicos sin Fronteras**, de la Figura 42, tenemos la relación recursiva “jefe” para la tabla “voluntarios”, pero en esta oportunidad, se resuelve creando una nueva tabla por la relación recursiva.

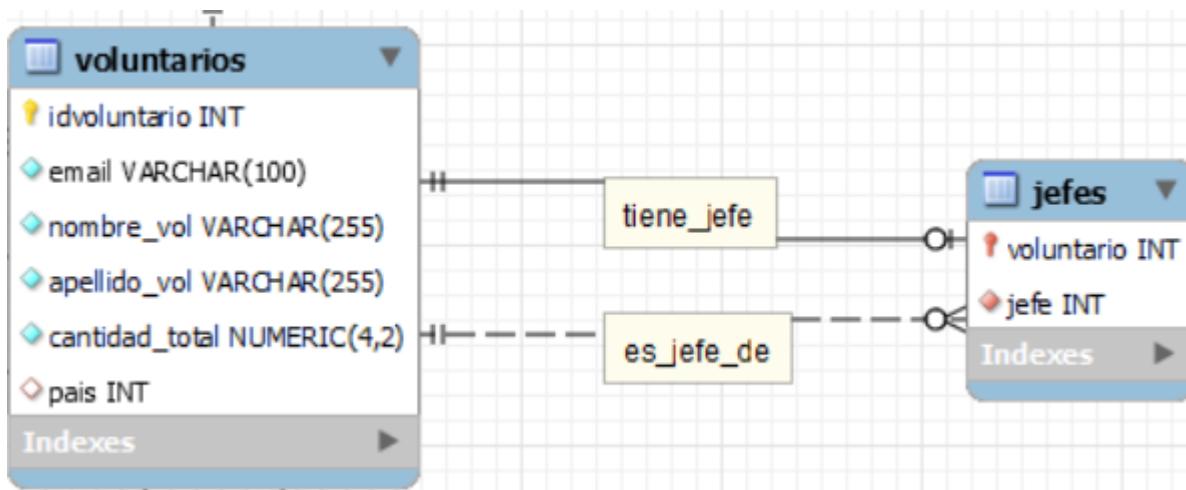


Figura 42

Entidades débiles por identificación

Una entidad débil se define así porque necesita de otra entidad (fuerte) para existir, una entidad débil siempre lo es por existencia, pero puede ser o no débil por identificación, dependiendo de si tiene o no un atributo identificador propio. En el caso de que no tenga identificador propio necesitará de la entidad fuerte para identificarse.

Cuando una entidad sólo es débil por existencia, en el esquema relacional se resuelve como cualquier otra tabla, agregando PK subrogada y definiendo como CK simples a los atributos identificadores de la entidad en el esquema lógico. Sin embargo, cuando es débil por identificación, si bien también la resolveremos como cualquier tabla, agregando la PK subrogada, deberemos tener en cuenta la definición de la clave candidata compuesta en la tabla, de manera que haya trazabilidad entre los esquemas lógico y relacional.

En el **Caso de Estudio de Médicos sin Fronteras**, en el Diagrama de Entidad - Relación (DER) del esquema lógico, tenemos como entidad débil por identificación a la entidad “alojamientos”, como se visualiza en la Figura 43.

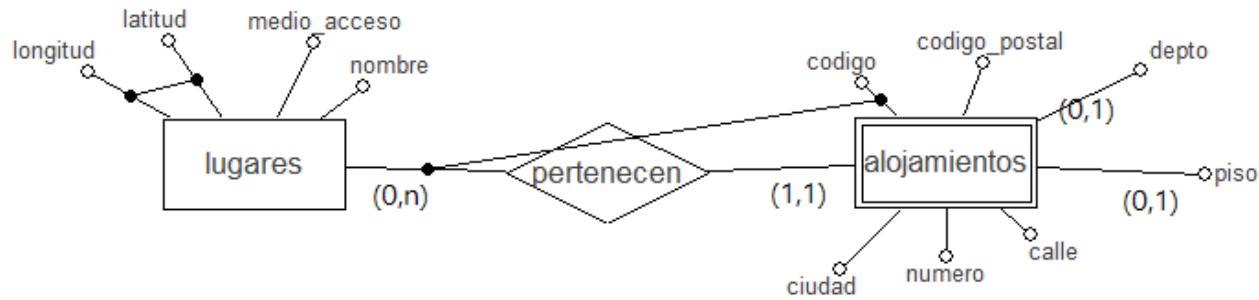


Figura 43

En el esquema relacional como primer paso, crearemos la tabla “alojamientos” la cual tendrá como PK subrogada a la columna “idalojamiento”, como puede verse en la Figura 44.

alojamientos	
!	idalojamiento INT
◆	codigo VARCHAR(45)
◆	calle VARCHAR(255)
◆	numero INT
◆	ciudad VARCHAR(255)
◆	codigo_postal VARCHAR(5)
◆	piso INT
◆	depto VARCHAR(45)
◆	lugar INT
Indexes	
	PRIMARY
	fk_idx_alojamientos_lugares
	ck_alojamientos

a) Tabla alojamiento



alojamientos - Table

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idalojamiento	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>					
codigo	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
calle	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					
numero	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>					

Column Name: **idalojamiento**

Charset/Collation: Default Charset Default Collation

Comments:

Data Type: INT
Default:
Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

b) Columnas de la tabla alojamiento donde puede verse la PK subrogada idalojamiento

Figura 44

En el esquema lógico, vemos que la entidad alojamiento tiene un identificador externo compuesto por el atributo **código** y el atributo identificador de la tabla “lugares”. Además, ambas entidades se relacionan por medio de la relación con conectividad 1:N “pertenecen” (Figura 43).

Para realizar la transformación en el esquema relacional, creamos la tabla alojamientos como se ve en la Figura 44 y resolvemos la relación “pertenecen” con la tabla “lugares” como se observa en la Figura 45.

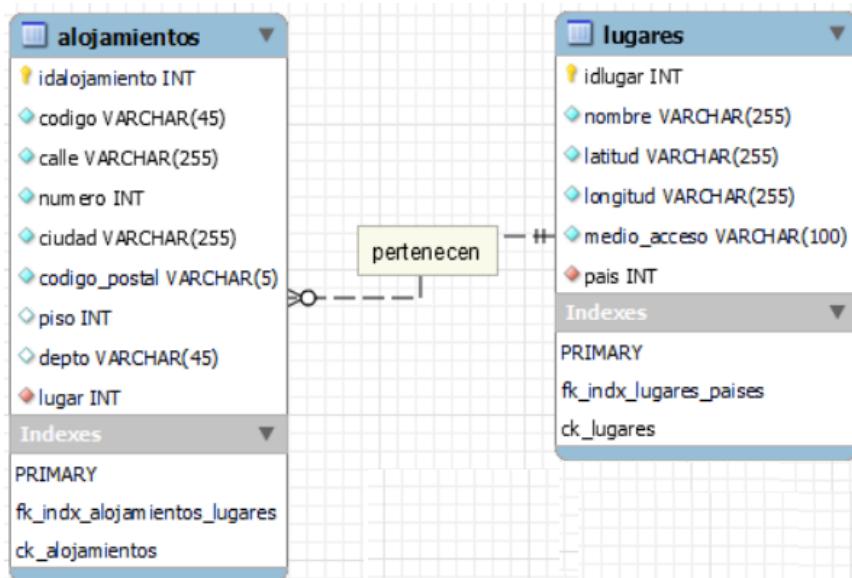
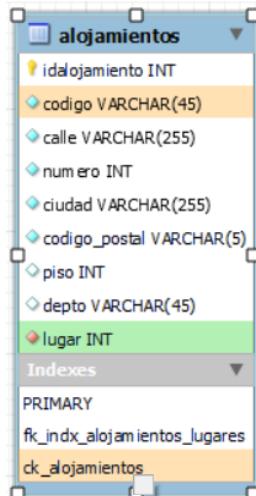


Figura 45

Luego, debemos definir la CK compuesta para la tabla “alojamientos”, que se compone de la columna lugar que es una FK a la tabla “lugares” y la columna código, como se observa en la Figura 46.



a) Ck compuesta de la tabla alojamiento



alojamientos - Table

Table Name: alojamientos Schema: ong

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idalojamiento	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
codigo	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
lugar	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
calle	VARCHAR(255)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name: lugar

Charset/Collation: Default Charset Default Collation

Comments:

Data Type: INT
Default:
Storage: Virtual Stored
Primary Key Not Null Unique
Binary Unsigned Zero Fill
Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

b) Las columnas código y lugar son NN

alojamientos - Table

Table Name: alojamientos Schema: ong

Index Name	Type
PRIMARY	PRIMARY
fk_idx_alojamientos...	INDEX
ck_alojamientos	UNIQUE

Index Columns

Column	#	Order	Length
idalojamiento		ASC	
<input checked="" type="checkbox"/> codigo	1	ASC	
<input type="checkbox"/> calle		ASC	
<input type="checkbox"/> numero		ASC	
<input type="checkbox"/> ciudad		ASC	
<input type="checkbox"/> codigo_postal		ASC	
<input type="checkbox"/> piso		ASC	
<input type="checkbox"/> depto		ASC	
<input checked="" type="checkbox"/> lugar	2	ASC	

Index Options
Storage Type:
Key Block Size: 0
Parser:
Visible:
Index Comment:

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

c) Se crea el índice UNIQUE para las columnas código y lugar, que darán lugar a la CK compuesta de la tabla alojamiento

Figura 46

Agregaciones

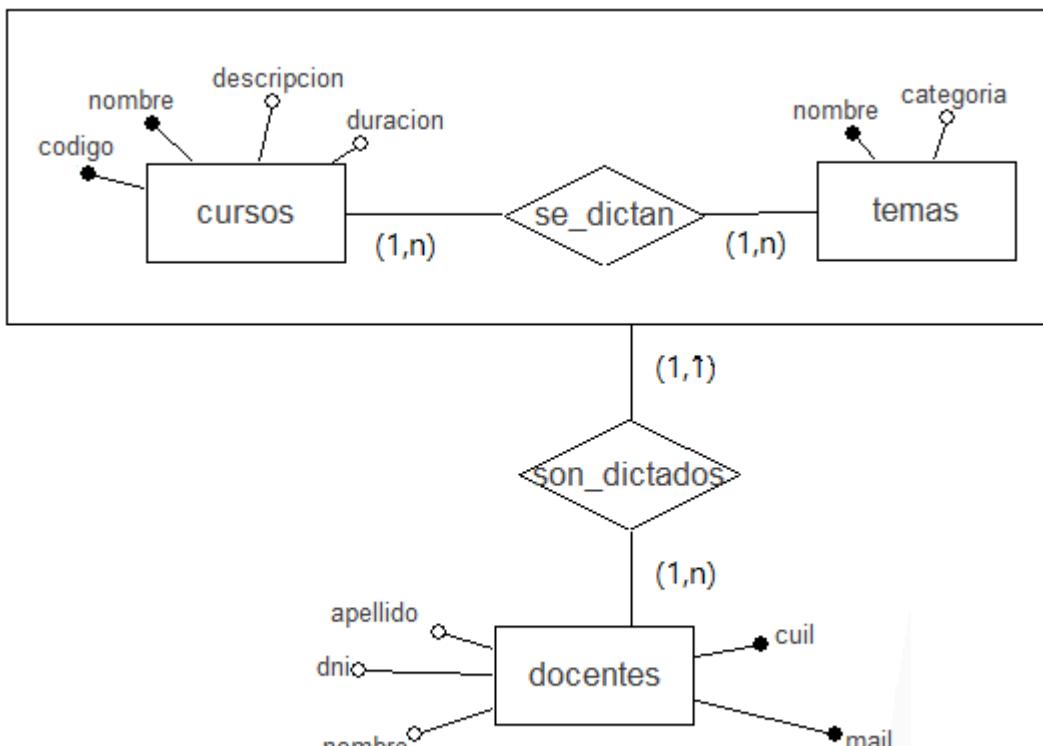
Para resolver una agregación, en primer lugar se resuelve la relación con conectividad N:M, como ya se explicó en apartados anteriores, y, luego se resuelve la relación 1:N de la misma forma que se explicó con anterioridad, con la nueva tabla resultante de la relación N:M.



En el siguiente ejemplo para cursos de posgrado, tenemos las siguientes reglas de negocio:

1. Los cursos de posgrado tienen un código y un nombre, ambos son únicos. Además, tienen una duración en meses y una breve descripción del campo del saber al cual corresponden.
2. En los cursos se dictan temas. Cada tema tiene un nombre que es único y una categoría que toma los valores principal o secundario.
3. Los temas en los cursos son dictados por docentes, sabiendo que un curso se dictan muchos temas, pero un tema en un curso es dictado por una o un solo docente. Pero un docente puede dictar uno o más temas en los distintos cursos.
4. De cada docente se almacena el cuil, dni, mail, nombre y apellido.

En la Figura 47, se visualiza el DER del esquema lógico correspondiente al ejemplo dado. Vemos que se resuelve con el concepto de **agregación**.



categoria = {principal, secundario}

Figura 47

Para resolver este esquema lógico en el esquema relacional usando la herramienta MySQL WorkBench, como ya hemos visto con anterioridad, primero debemos crear las tres tablas y luego resolver las dos relaciones.

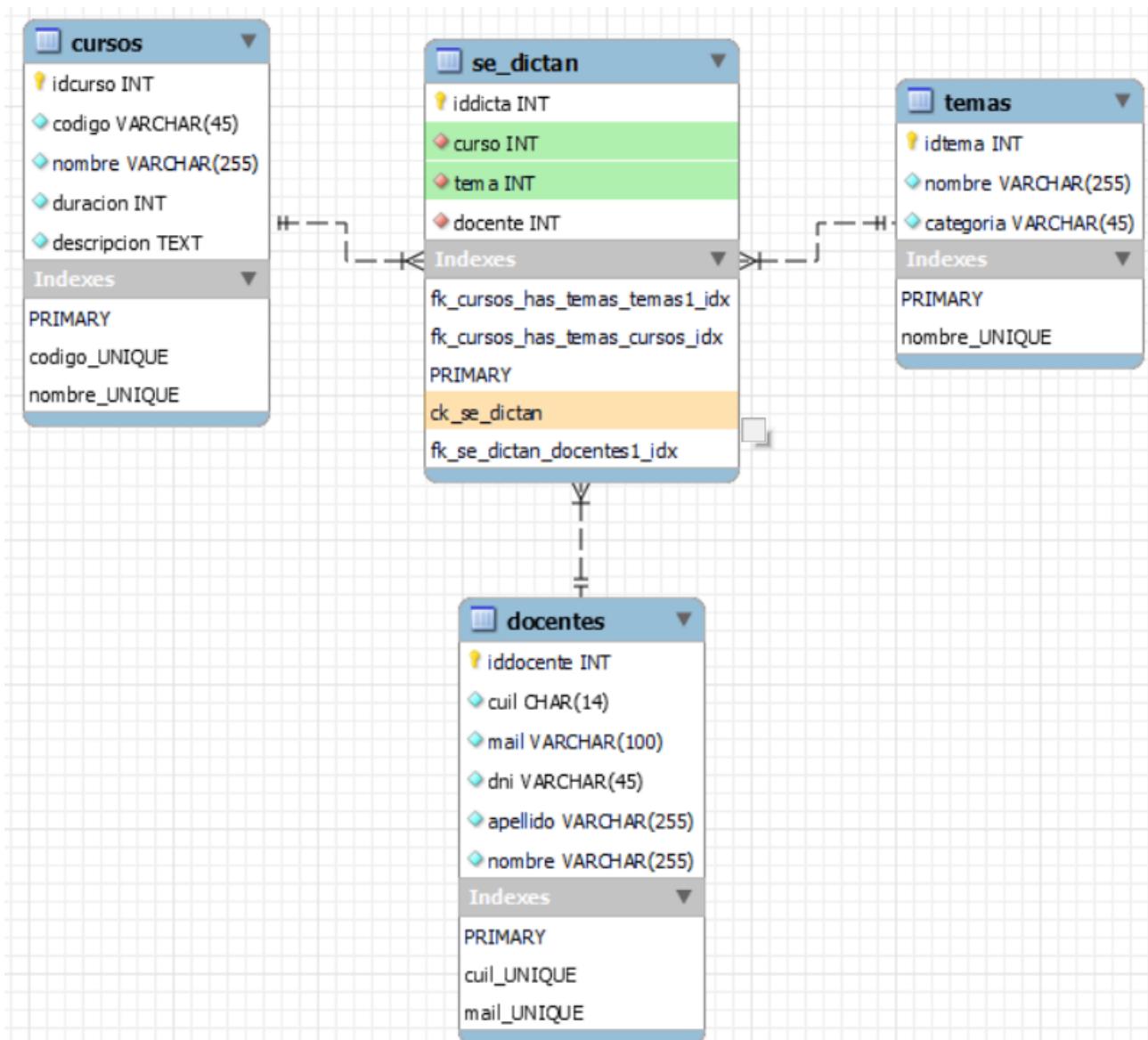


Figura 48

Como se puede observar en la Figura 48, se encuentran creadas las tres tablas “cursos”, “temas” y “docentes”. Luego, se creó la tabla “se_dictan” correspondiente a la relación N:N y por último, se resuelve la relación 1:N entre la tabla “se_dictan” y la tabla “docentes”.



Como puede observarse en la Figura 49 la CK compuesta de esta tabla corresponde a las columnas “curso” y “tema”, resultantes de la relación N:M. La columna “docente” no forma parte de la misma.

Index Name	Type
fk_cursos_has_temas...	INDEX
fk_cursos_has_temas...	INDEX
PRIMARY	PRIMARY
ck_se_dictan	UNIQUE
fk_se_dictan_docente...	INDEX

Column	#	Order	Length
<input type="checkbox"/> iddicta		ASC	
<input checked="" type="checkbox"/> curso	1	ASC	
<input checked="" type="checkbox"/> tema	2	ASC	
<input type="checkbox"/> docente		ASC	

Figura 49

Relaciones Ternarias

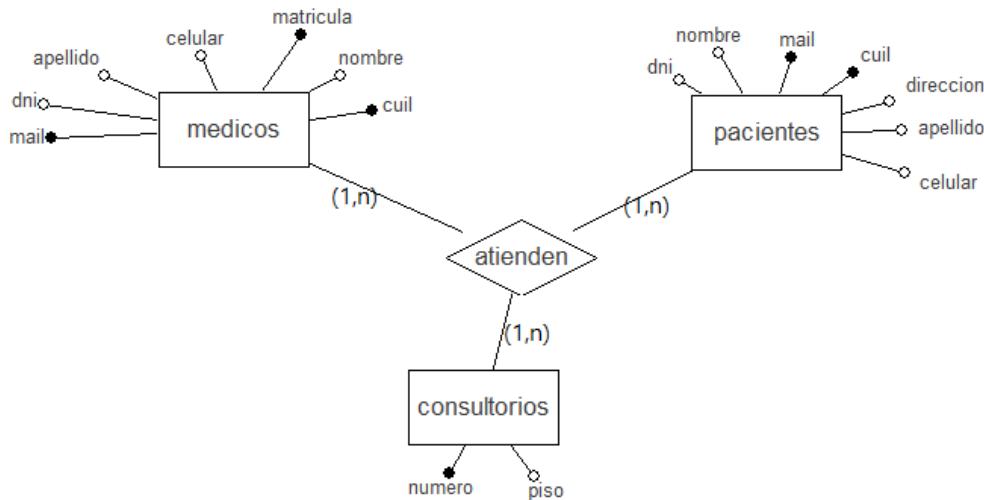
Para resolver una relación ternaria, en primer lugar se crea una relación con conectividad N:M, entre dos de las entidades, como ya se explicó en apartados anteriores, y, luego se resuelve la relación 1:N de la misma forma que se explicó con anterioridad, con la nueva tabla resultante de la relación N:M y la tabla que aún no fue relacionada.

En el siguiente ejemplo para las consultas ambulatorias de una Clínica, tenemos las siguientes reglas de negocio:

1. De los médicos se sabe su matrícula, cuil y mail, que son únicos, dni, nombre, apellido y celular.
2. De los pacientes se sabe su cuil y mail, que son únicos, dni, nombre, apellido , dirección y celular.



3. De los consultorios se sabe el número que es único y el piso en que se ubica dentro de la clínica. Hay cinco pisos.
 4. Los médicos atienden a los pacientes en los distintos consultorios.



piso = [1,5] números enteros

Figura 50

En la Figura 50 puede verse el DER del esquema lógico para el ejemplo dado. Vemos que se resuelve con el concepto de **relación ternaria**.

Para resolver este ejemplo usando la herramienta MySQL WorkBench, primero debemos crear las tres tablas para “medicos”, “pacientes” y “consultorios” y, luego resolver la relación ternaria en dos pasos. En el primer paso se creará la relación N:M “atienden” entre las tablas “medicos” y “pacientes”. En el segundo paso, se relacionarán por medio de una relación con conectividad 1:N no identificatoria, a las tablas “atienden” y “consultorios”. Esto puede observarse en la Figura 51.

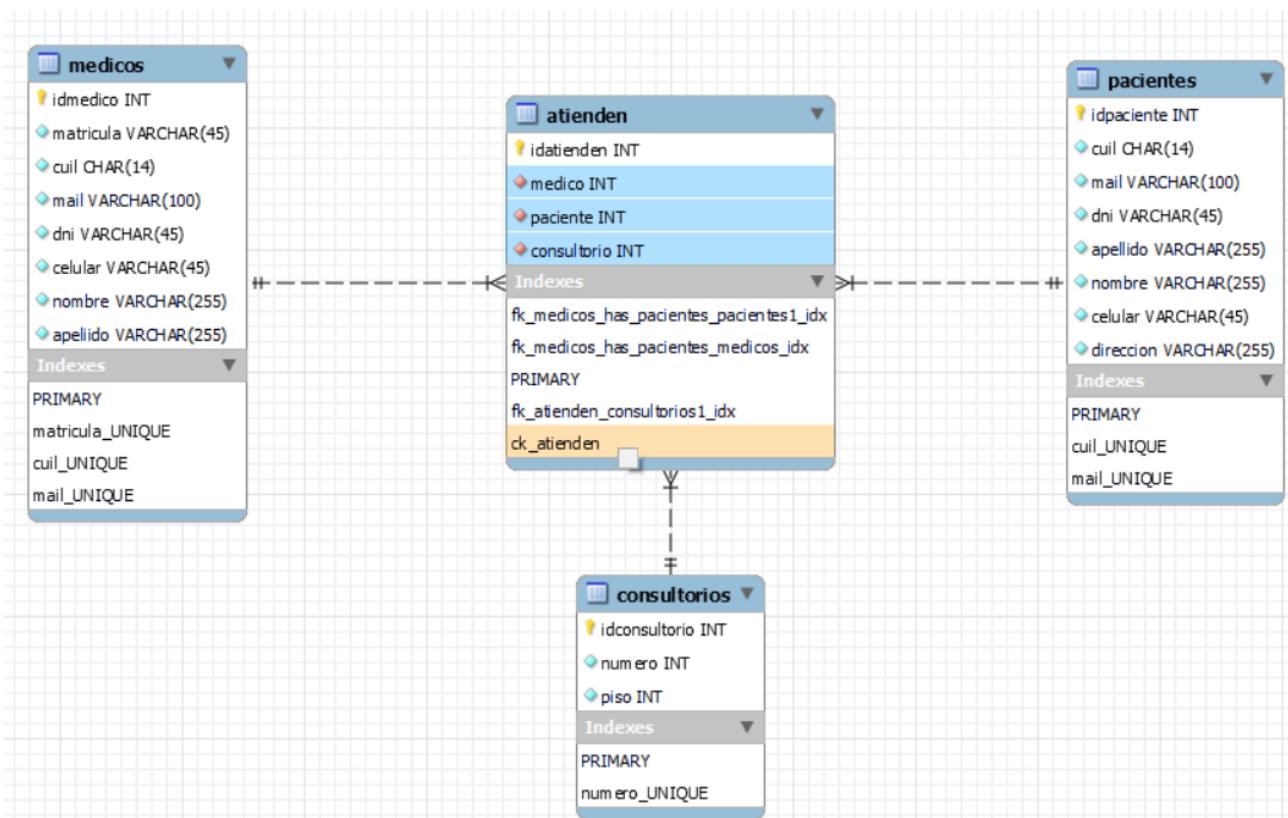


Figura 51

A continuación, se creará la CK compuesta para la tabla “atienden” como se muestra en la Figura 52.



atienden - Table X

Table Name:		atienden	Schema: mydb		
Index Name	Type				
fk_medicos_has_pac...	INDEX				
fk_medicos_has_pac...	INDEX				
PRIMARY	PRIMARY				
fk_atienden_consulto...	INDEX				
ck_atienden	UNIQUE				

Index Columns

Column	#	Order	Length
<input type="checkbox"/> idatienden		ASC	
<input checked="" type="checkbox"/> medico	1	ASC	
<input checked="" type="checkbox"/> paciente	2	ASC	
<input checked="" type="checkbox"/> consultorio	3	ASC	

Index Options

- Storage Type:
- Key Block Size: 0
- Parser:
- Visible:

Index Comment:

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

Figura 52

Historial

Un historial es una relación binaria (puede ser ternaria) que introduce el concepto de tiempo en el esquema conceptual, para resolverla debemos tener en cuenta los atributos polivalentes de la relación correspondientes a fechas o períodos de tiempo.

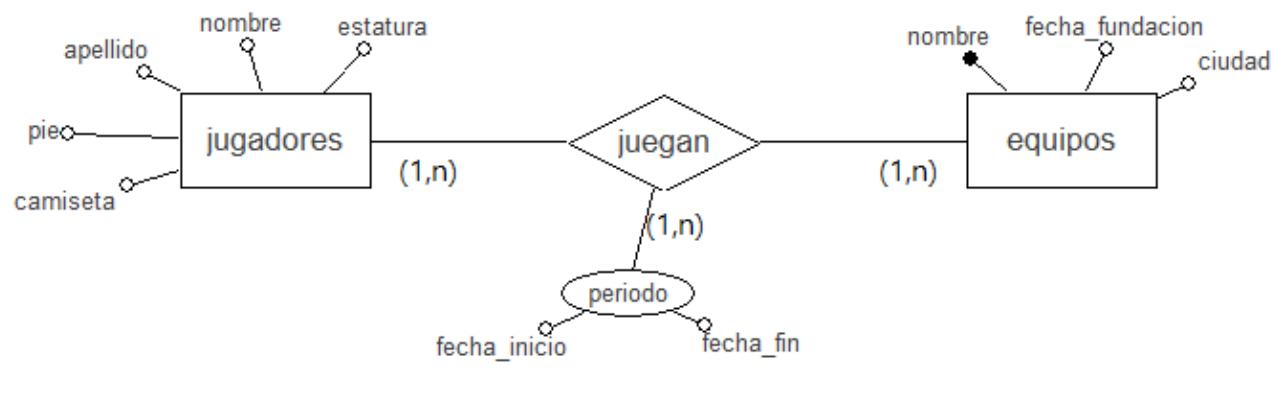
En el siguiente ejemplo sobre los equipos en los que juegan los jugadores de fútbol, tenemos las siguientes reglas de negocio:

1. De los jugadores de fútbol se sabe el nombre, apellido, estatura, con que pie juega (derecho, izquierdo) y se sabe el número de camiseta que usan, que se puede repetir entre los distintos jugadores.
2. De los equipos se sabe el nombre, que es único, la fecha de fundación y la ciudad a la que pertenecen.
3. Los jugadores juegan en los distintos equipos por diferentes períodos de tiempo, con una fecha de inicio y una fecha de fin.



En la Figura 53, se visualiza el DER del esquema conceptual correspondiente al ejemplo dado. Vemos que se resuelve con el concepto de **historial**.

Esquema conceptual



pie = {derecho, izquierdo}

Figura 53

En la Figura 54, se visualiza el DER del esquema lógico correspondiente al ejemplo dado. Vemos que se resuelve el atributo compuesto, pero no así, los atributos polivalentes, ya que esto nos permitirá recordar que formarán parte de la CK de la tabla resultante de resolver la relación “juegan” con conectividad N:M, pero en el esquema relacional serán atributos simples.

Esquema lógico

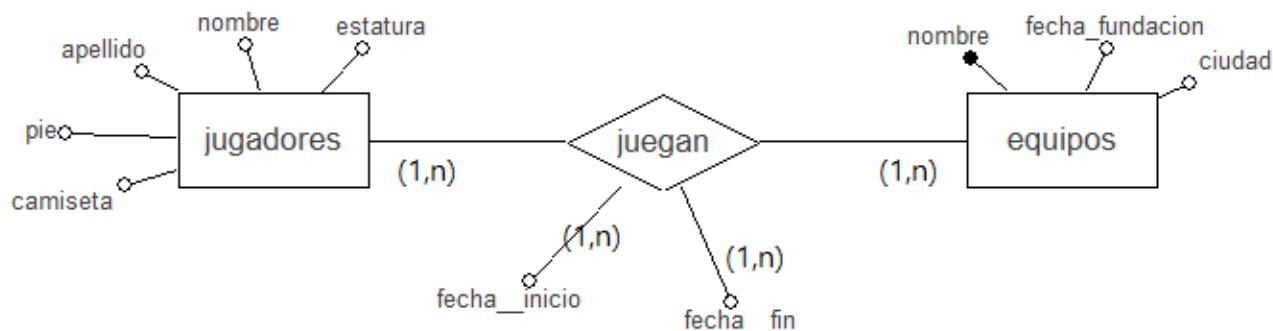


Figura 54

Para resolver este ejemplo usando la herramienta MySQL WorkBench, primero debemos crear las tablas “jugadores” y “equipos”, para luego resolver la relación N:M “juegan” entre ambas tablas. Esto se muestra en la Figura 55.

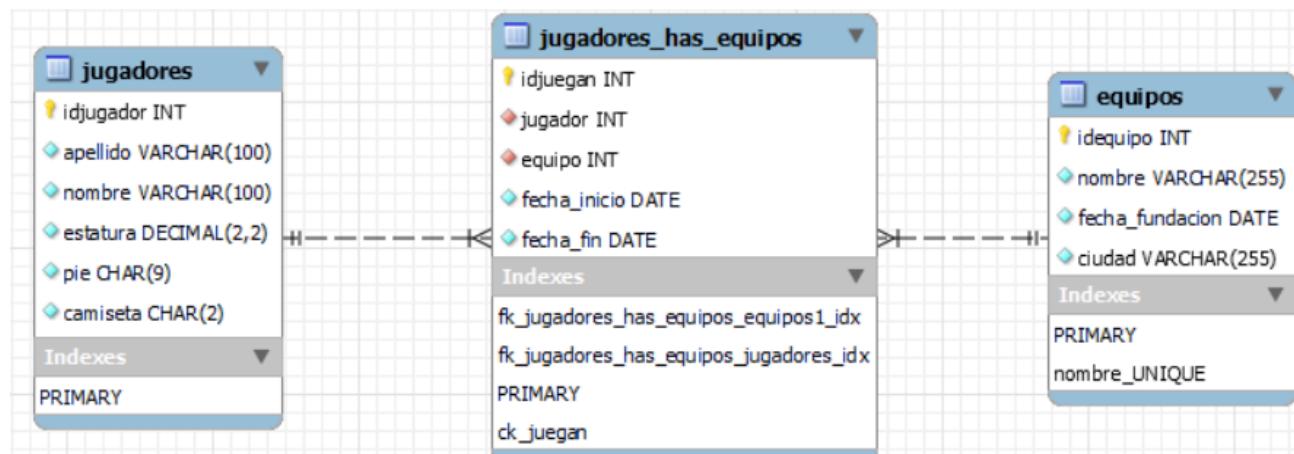


Figura 55



La CK compuesta de la tabla se compone de las columnas “jugador”, “equipo” y “fecha_inicio” conjuntamente, ya que, hay que permitir la posibilidad de que un jugador juegue en más de un equipo en distintos períodos de tiempo, e incluso, juegue en el mismo equipo en más de un período, por esto último, la fecha de inicio forma parte de la CK compuesta. Esto puede verse en la Figura 56.

Column	#	Order	Length
<input type="checkbox"/> idjuegan		ASC	
<input checked="" type="checkbox"/> jugador	1	ASC	
<input checked="" type="checkbox"/> equipo	2	ASC	
<input checked="" type="checkbox"/> fecha_inicio	3	ASC	
<input type="checkbox"/> fecha_fin		ASC	

Figura 56

Esquema Físico

Ingeniería hacia adelante

La ingeniería hacia adelante nos permite, dado un esquema relacional de una base de datos, obtener el esquema físico de la misma en un SGBD, en este caso, usaremos el SGBD MySQL Server 5.5, el cual debe estar instalado con anterioridad en nuestra computadora de forma local o debemos tener los datos para conectarnos al mismo de forma remota (ver Figura 59).



El esquema relacional de la Figura 57 corresponde a la base de datos llamada “ong” del **Caso de Estudio de Médicos sin Fronteras**, y es la que usaremos para el ejemplo de este apartado.

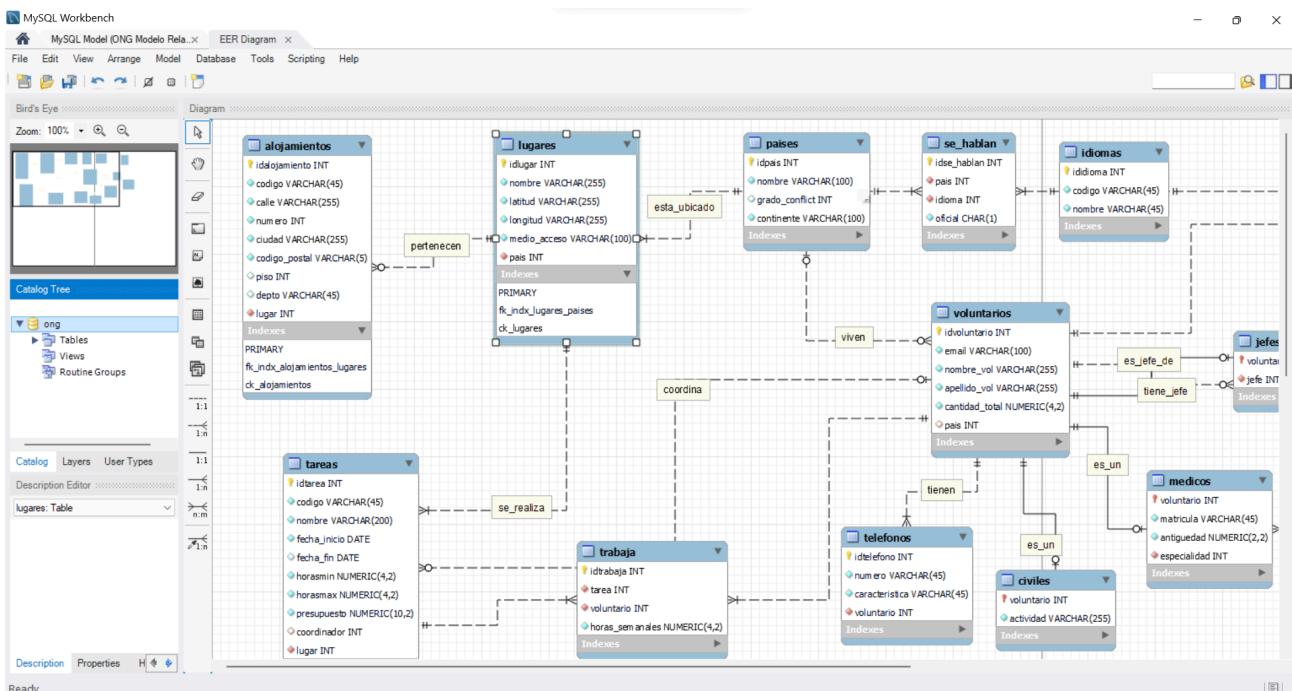


Figura 57

Una vez que se ha creado el esquema relacional en MySQL Workbench, y se le ha cambiado el nombre al esquema cambiando el que viene por defecto (Ver apartado anterior *Cambiar el nombre del esquema de la base de datos*). Se procede a realizar la ingeniería hacia adelante desde el menú superior “Database”, luego “Forward Engineer” como se muestra en la Figura 58.

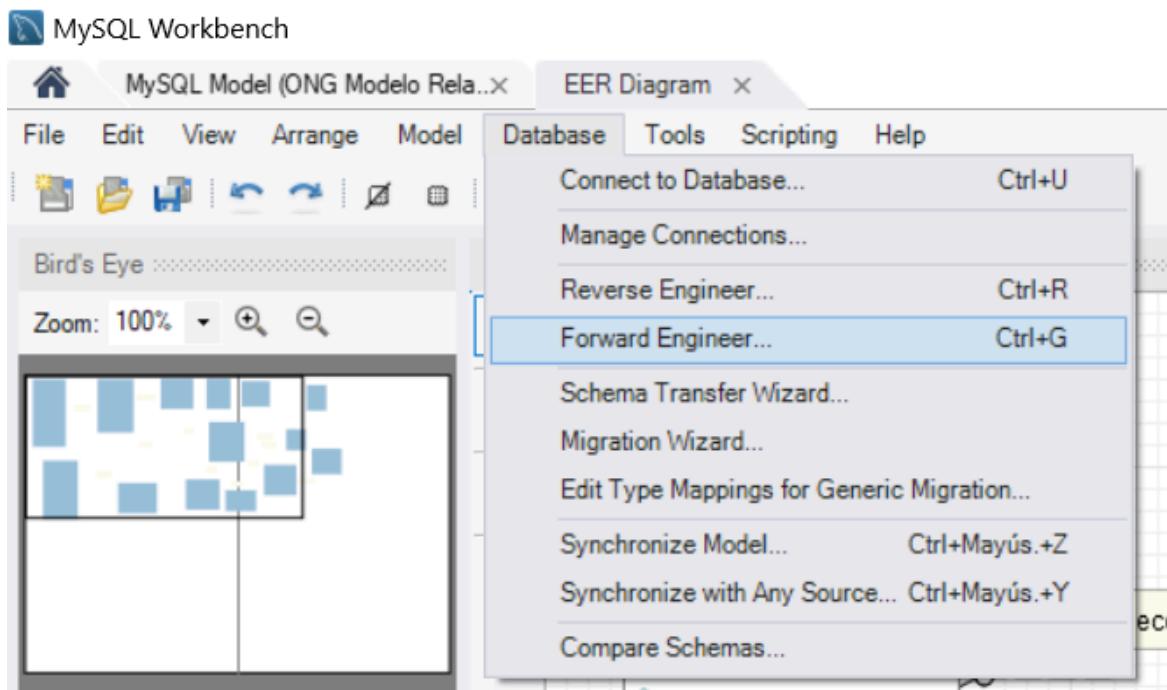


Figura 58

Esta acción abre la ventana que se visualiza en la Figura 59. Aquí debemos conectarnos al servidor de base de datos MySQL Server, en este caso, está instalado como una instancia local, por el puerto por defecto 3306 y nos conectaremos por medio del usuario root. Se solicitará la password si no es que está almacenada como es este caso de ejemplo (*No debería ser así por cuestiones de seguridad*).

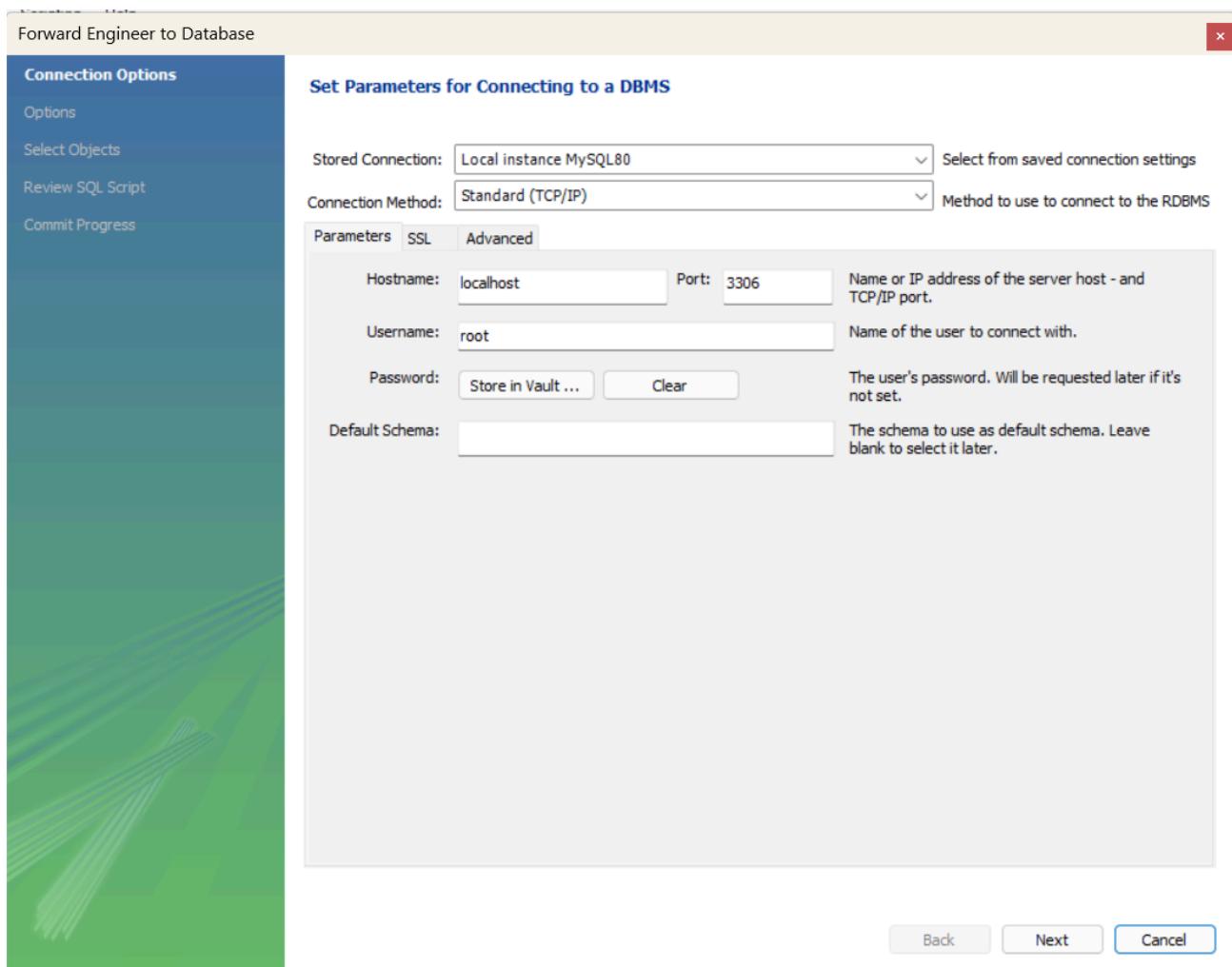


Figura 59

Haciendo click en el botón “Next” se abre la pantalla que se visualiza en la Figura 60.

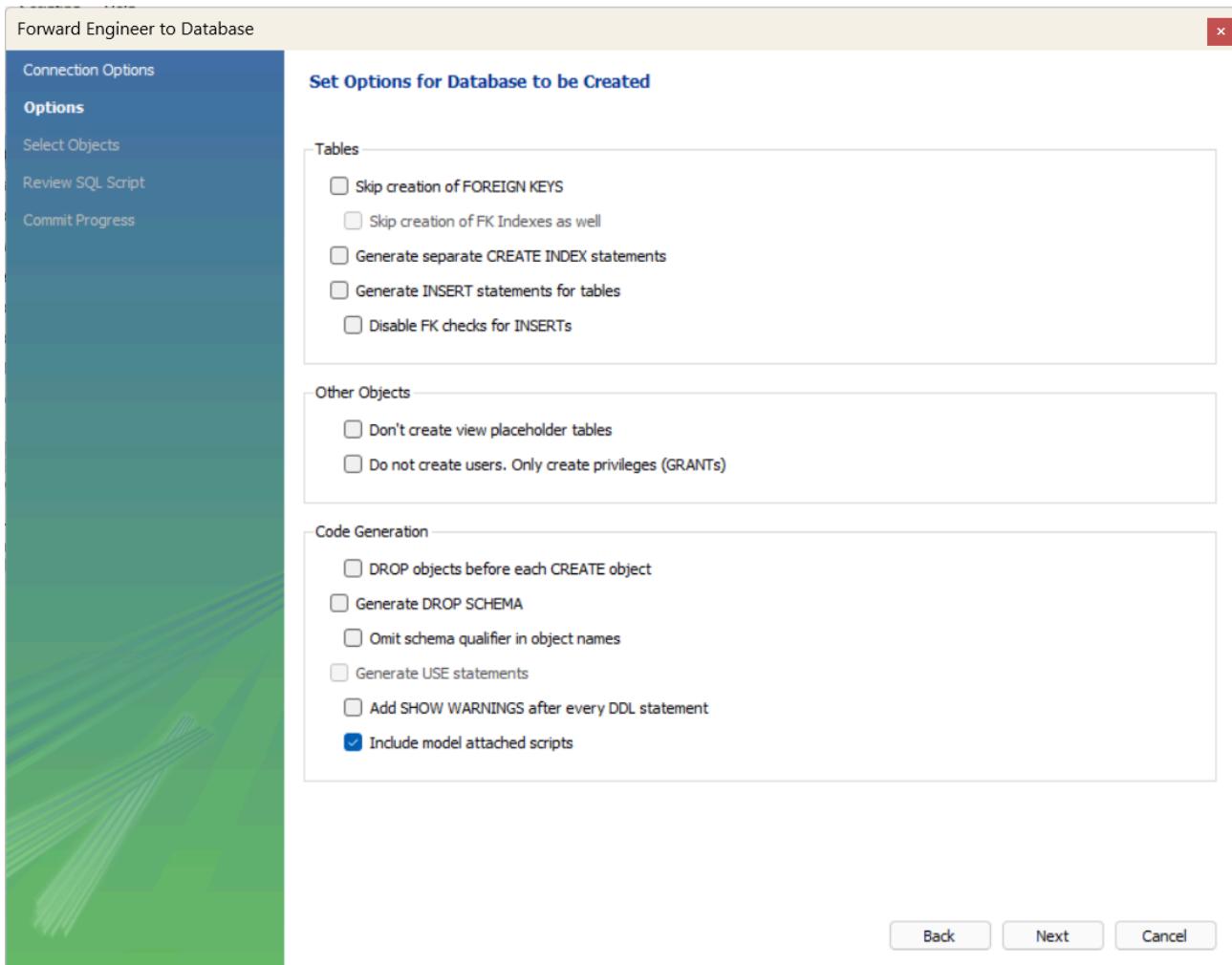


Figura 60

En esta ventana se deja todo como aparece por defecto y se hace click en el botón “Next”, acción que nos conduce a la siguiente ventana que se muestra en la Figura 61.

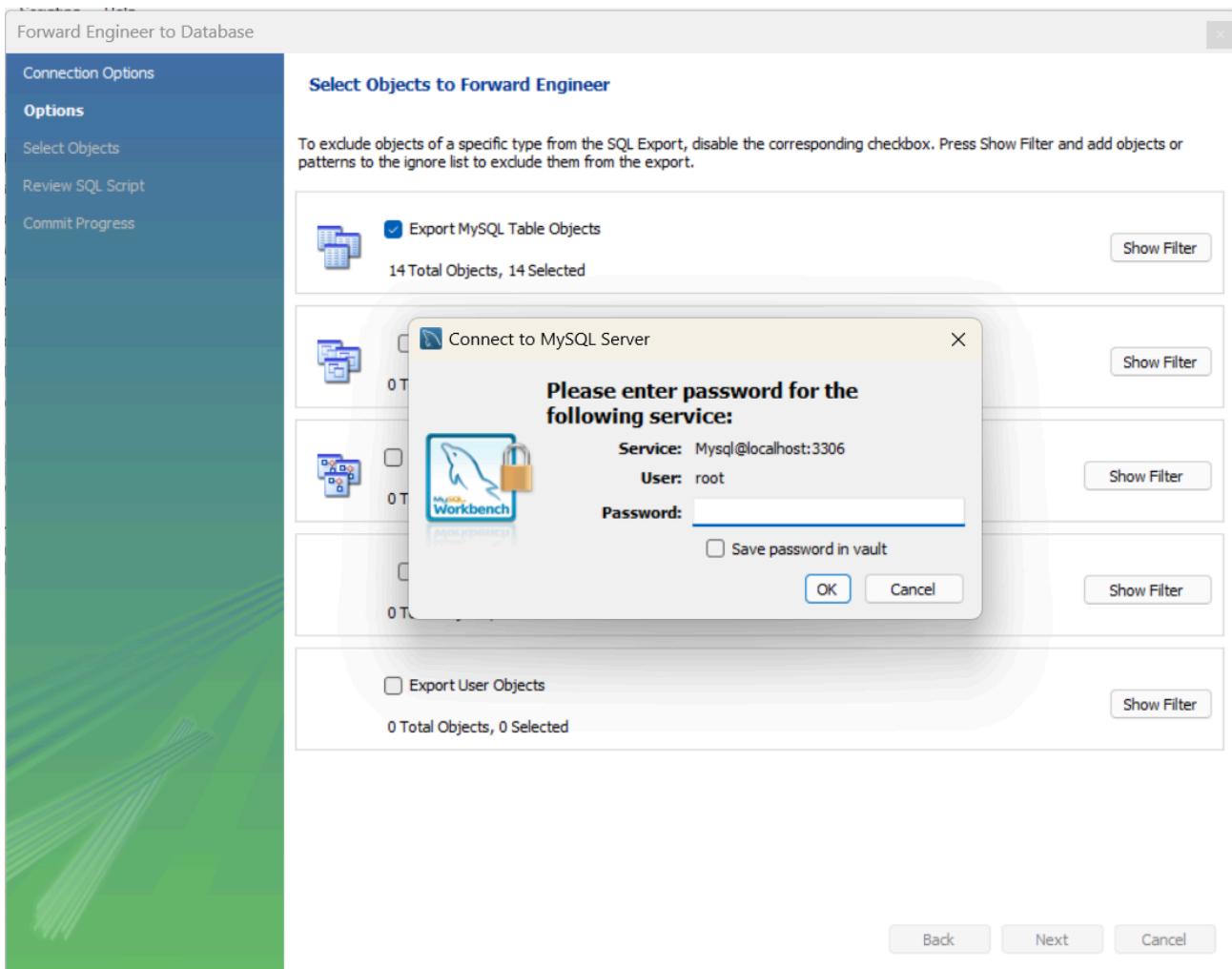


Figura 61

Nuevamente se nos solicita la password del usuario de MySQL Server, la escribimos y luego hacemos click en el botón “OK”. Esta acción nos conduce a la ventana que se visualiza en la Figura 62. Allí controlamos y tildamos los objetos que queremos en la base de datos sean correctos, en este caso solo las tablas, sus restricciones e índices.



Por último, hacemos click en el botón “Next”, lo cual nos conduce a la ventana que se ve en la Figura 63. Allí podemos ver que se visualiza el lenguaje SQL DDL para la creación de los objetos de la base de datos en el esquema físico, y hacemos nuevamente click en el botón “Next”. Nuevamente se nos solicita la password del usuario de MySQL Server (Figura 64). La completamos y luego se abre la ventana de la Figura 65, donde vemos que automáticamente se conecta al servidor, ejecuta el script (SQL DDL) de creación del esquema físico y salvaguarda los datos. Hacemos click en el botón “Close”.

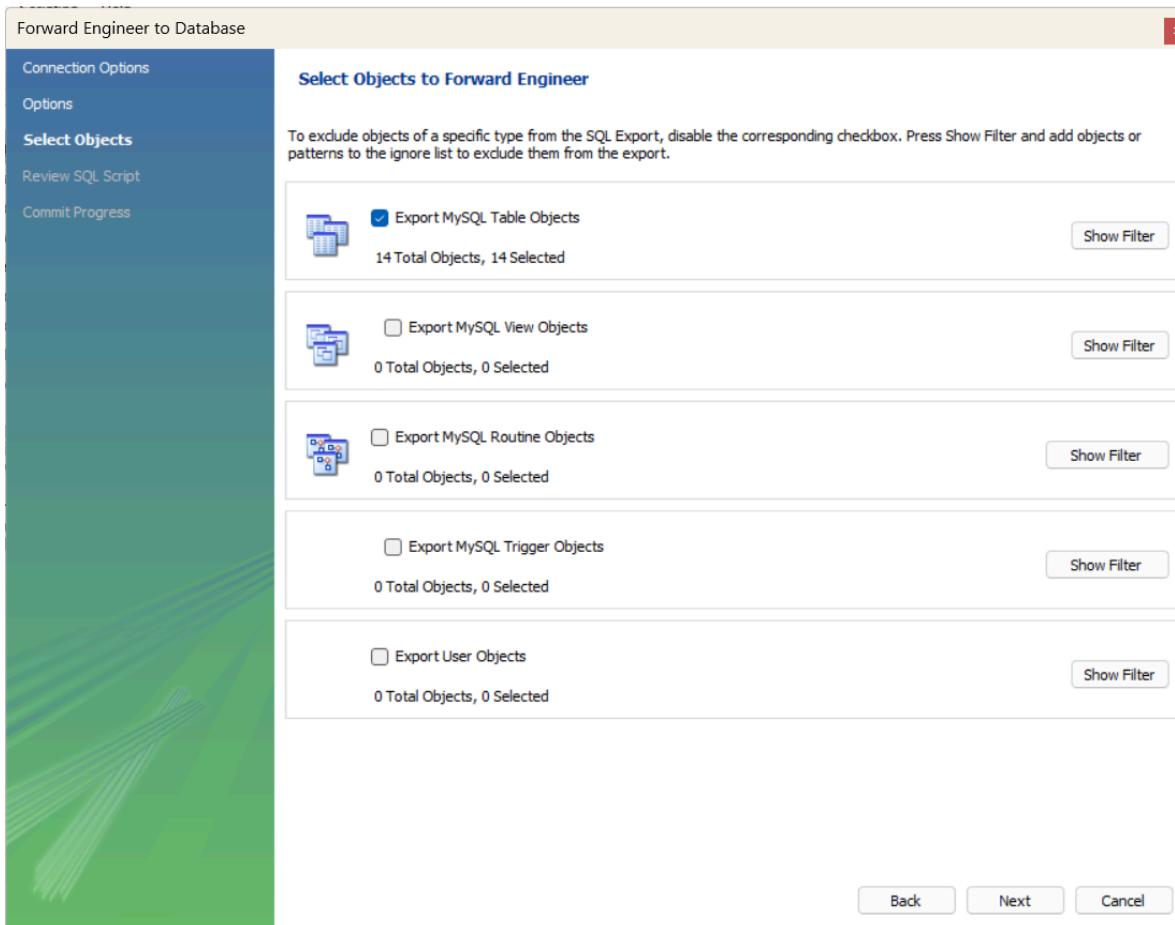


Figura 62



Forward Engineer to Database

Connection Options
Options
Select Objects
Review SQL Script
Commit Progress

Review the SQL Script to be Executed

This script will now be executed on the DB server to create your databases.
You may make changes before executing.

```
1 -- MySQL Workbench Forward Engineering
2
3 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLE!
6
7 -----
8 -- Schema ong
9 -----
10 -----
11 -----
12 -- Schema ong
13 -----
14 CREATE SCHEMA IF NOT EXISTS `ong` DEFAULT CHARACTER SET utf8 ;
15 USE `ong` ;
16
17 -----
18 -- Table `ong`.`paises`
19 -----
20 CREATE TABLE IF NOT EXISTS `ong`.`paises` (
21     `idpais` INT NOT NULL AUTO_INCREMENT,
22     `nombre` VARCHAR(100) NOT NULL,
23     `grado_conflict` INT NULL DEFAULT 0,
24     `continente` VARCHAR(100) NOT NULL,
```

Save to File... Copy to Clipboard

Back Next Cancel

Figura 63

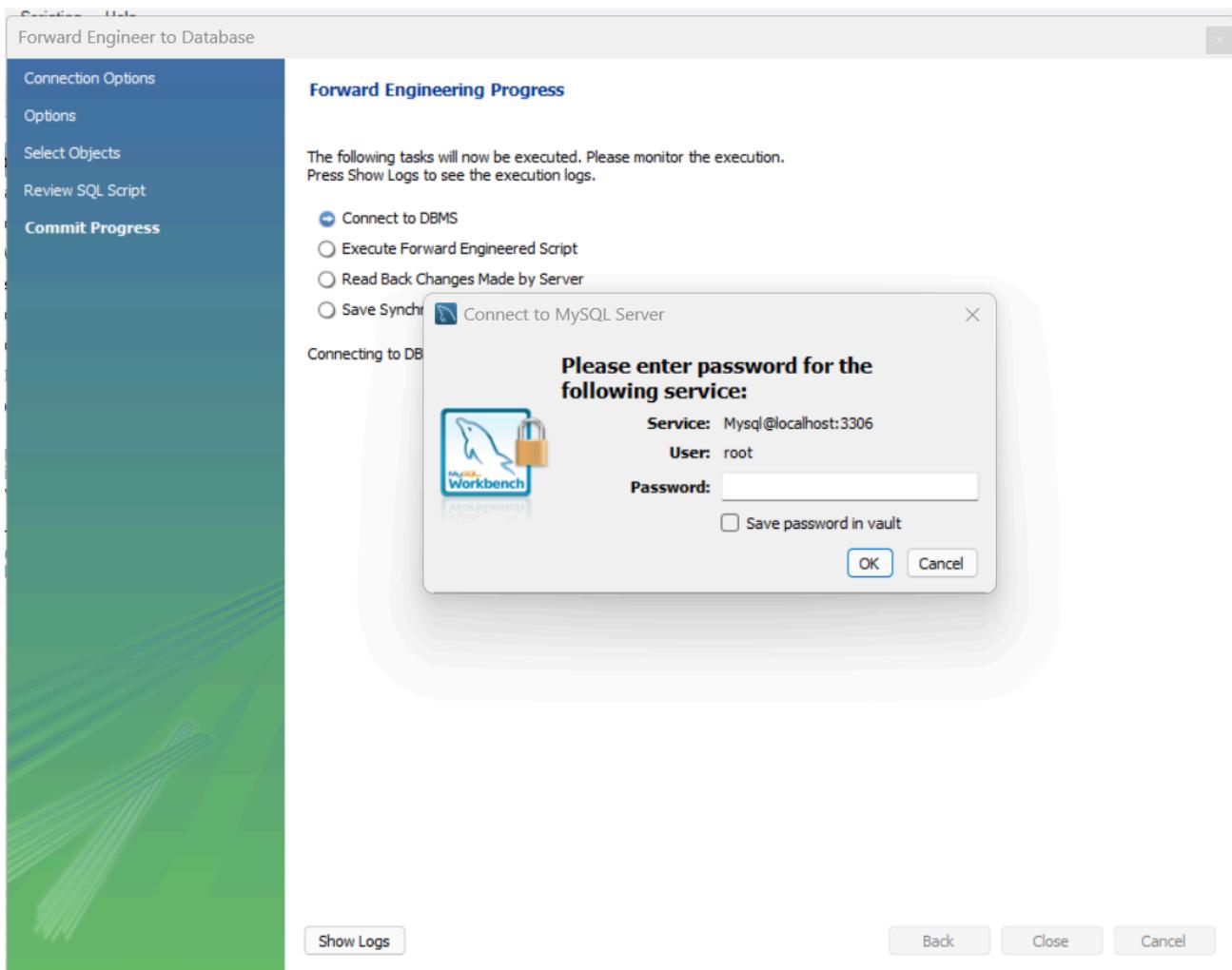


Figura 64

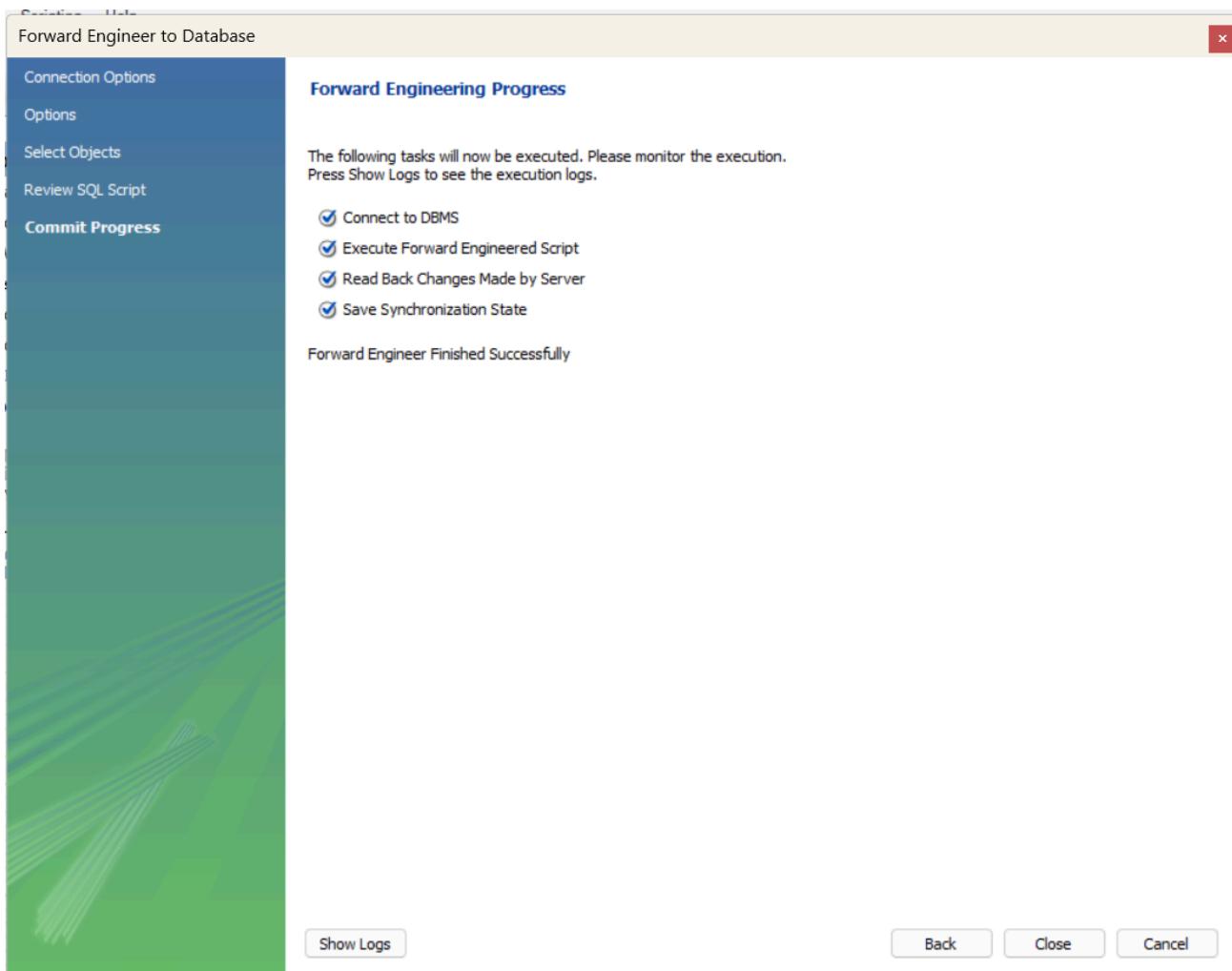


Figura 65



Visualizar el esquema físico de la base de datos

Para visualizar el esquema físico de la base de datos recién creada debemos ir al home de MySQL WorkBench (Figura 66) y hacer click en la instancia del servidor MySQL en donde hayamos creado la base de datos, en este caso, solo tenemos una instancia local.

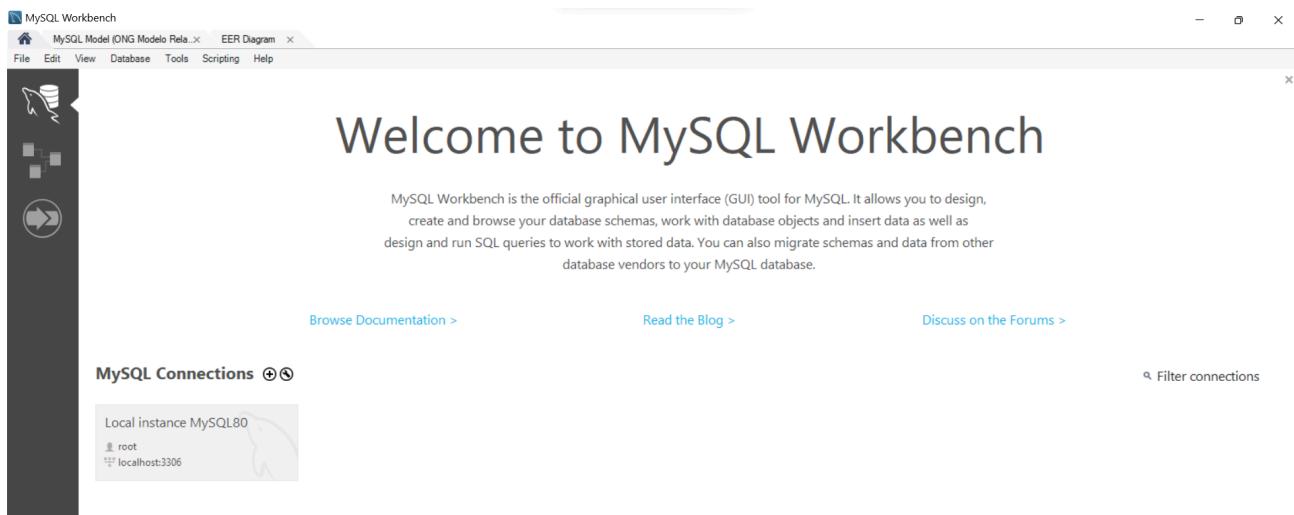


Figura 66

Una vez que ingresamos a la sección de la aplicación para gestionar las bases de datos, podemos trabajar con todas las bases creadas. En la Figura 67 se visualiza el esquema físico para la base de datos “ong” del **Caso de Estudio de Médicos sin Fronteras**. Ya podemos empezar a trabajar con nuestra nueva base de datos, además de la base de datos prueba y la base sys que viene por defecto.



MySQL Workbench

MySQL Model (ONG Modelo Rela.x) EER Diagram Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS MySQL Model (ONG Modelo Rela.x)

Tables

ong

alojamientos

- civiles
- especialidades
- hablan
- idiomas
- jefes
- lugares
- lenguajes
- paises
- se_hablan
- tareas
- telefonos
- trabaja
- voluntarios

Views

Stored Procedures

Functions

prueba

sys

Administration Schemas

Information

Table: alojamientos

Columns:

idalojamiento	int AI PK
codigo	varchar(4)
calle	varchar(2)
numero	int
ciudad	varchar(2)
codigo_postal	varchar(5)
piso	int
depto	varchar(4)
lugar	int

Action Output

#	Time	Action

Message

Duration / Fetch

Object Info Session

Figura 67



Anexo

Caso de Estudio Médicos sin Fronteras

Para este tutorial utilizamos el **Caso de Estudio de Médicos sin Fronteras**, el cual se presenta a continuación.

La ONG “Médicos sin Fronteras” tiene que manejar la información sobre las tareas que llevan a cabo sus voluntarios en las distintas zonas de conflicto distribuidas en diferentes lugares a lo largo del mundo. Se nos solicita que diseñemos la base de datos para el sistema informático de la ONG, para lo cual se tienen las siguientes reglas de negocio:

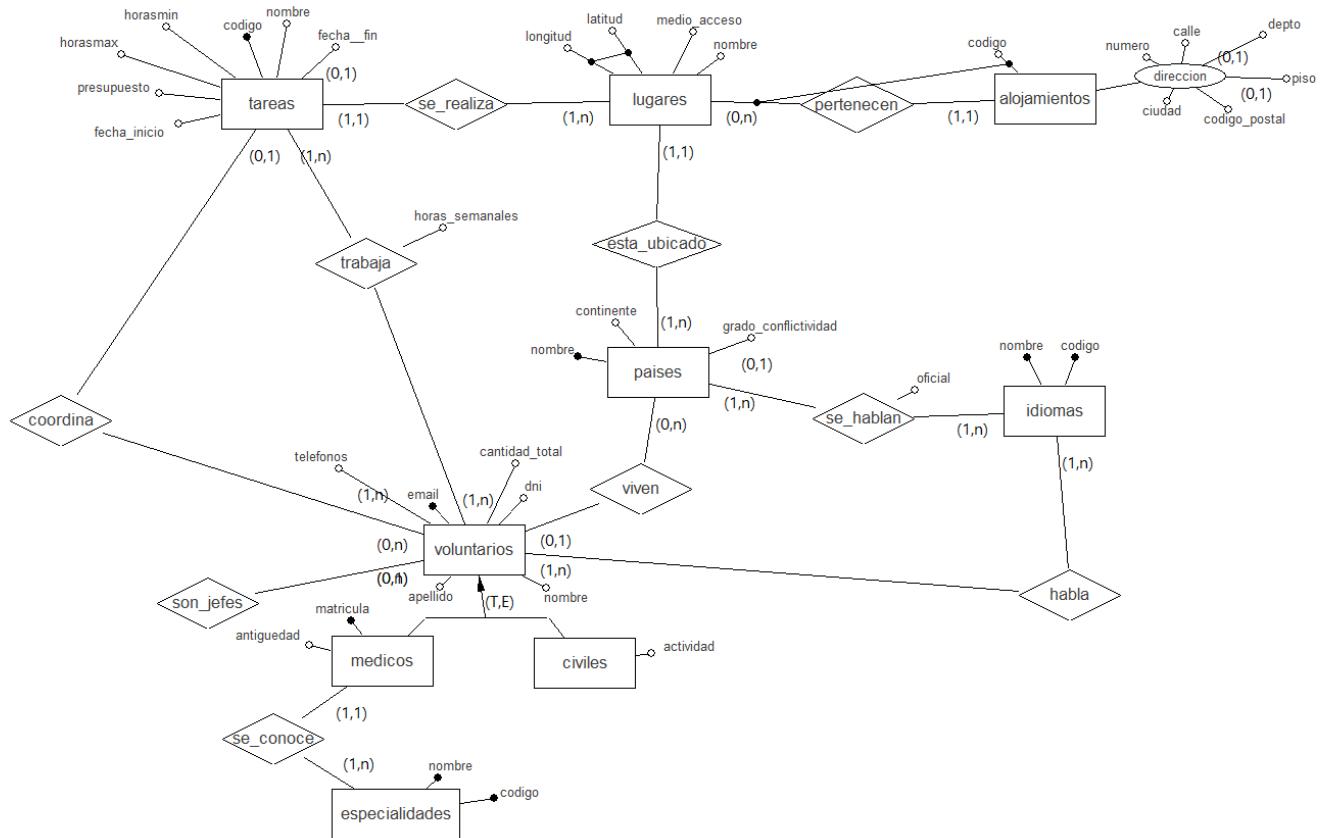
1. De cada tarea se sabe: el nombre (que puede repetirse entre las distintas tareas), el código de la tarea (que es único para cada tarea), cuántas horas por semana conviene dedicarle como mínimo, cuantas horas por semana se le puede dedicar como máximo, una fecha de inicio, una fecha de fin (si es que está finalizada) y el presupuesto asignado a la misma.
2. De los lugares donde se hacen tareas se debe manejar: un nombre que se puede repetir, sus coordenadas (latitud y longitud), que identifican al lugar, su medio de acceso (auto, 4x4, camión, avión, barco, tren), sus paradas para alojamiento (que pueden no tener) y se sabe además el país en donde está ubicado.
3. De cada alojamiento se almacena un código, que se repite dentro del lugar al que pertenecen y una dirección que está conformada por calle, número, ciudad, código postal, piso y depto, los dos últimos datos pueden no corresponder a una dirección.
4. Puede pasar que distintas tareas se hagan en el mismo lugar, pero no sucede que una misma tarea se realice en varios lugares. De hecho, cada tarea se realiza en un solo lugar.
5. De cada país se conoce su nombre que es único, el continente en que está ubicado (ASIA, AMÉRICA DEL SUR, AMÉRICA CENTRAL, AMÉRICA DEL NORTE, ÁFRICA, ANTÁRTIDA, EUROPA, OCEANÍA) y el grado de conflictividad (de



algunos países no se tiene esta información), que es un número entero del 1 al 10 (1: todo tranquilo, 10: guerra civil, etc.).

6. En los países se hablan distintos idiomas y cada idioma se habla al menos en un país. Cada país tiene como mínimo un idioma oficial, se quieren saber tanto los idiomas oficiales como los no oficiales.
7. De los idiomas se tiene un código alfanumérico único y el nombre, que también es único.
8. De cada voluntario de la ONG se sabe: nombre, apellido, email, teléfonos, DNI, que se puede repetir, país en donde vive y los idiomas que habla. Se toma como supuesto semántico que no puede haber dos personas que comparten su email, ya que es una forma de comunicarse con los voluntarios por parte de la ONG.
9. Los voluntarios pueden ser médicos o personas civiles que se alistan en la ONG para prestar distintos servicios. De los médicos se conoce su especialidad, la matrícula que es única y los años de antigüedad que hace que ejerce como tal. De las personas civiles se tiene una descripción de la actividad que puede aportar a la ONG. Por ejemplo, ayuda en tareas de enfermería, cocina, conduce autos y helicópteros, etc.
10. De cada especialidad se tiene un código y un nombre. Ambos datos deben ser únicos.
11. De algunos voluntarios se desconoce el país donde viven.
12. Cada voluntario trabaja en una o en muchas tareas, para cada una de ellas se sabe cuántas horas le dedica por semana. En una tarea pueden trabajar varios voluntarios.
13. Cada tarea puede tener (aunque no siempre) un coordinador, que es un voluntario, que puede o no trabajar en esa tarea. Un voluntario solo puede coordinar varias tareas.
14. Se quiere saber la cantidad total de horas semanales que trabaja un voluntario entre todas las tareas que realiza.
15. Algunos voluntarios son jefes de otro grupo de voluntarios.

En la Figura 68 se encuentra el Esquema Conceptual para el **Caso de Estudio de Médicos sin Fronteras**.



Dominios definidos por el usuario:

medio_acesso = {auto, 4x4, camión, avión, barco, tren}

continente = {ASIA, AMÉRICA DEL SUR, AMÉRICA CENTRAL, AMÉRICA DEL NORTE, ÁFRICA, ANTÁRTIDA, EUROPA, OCEANÍA}

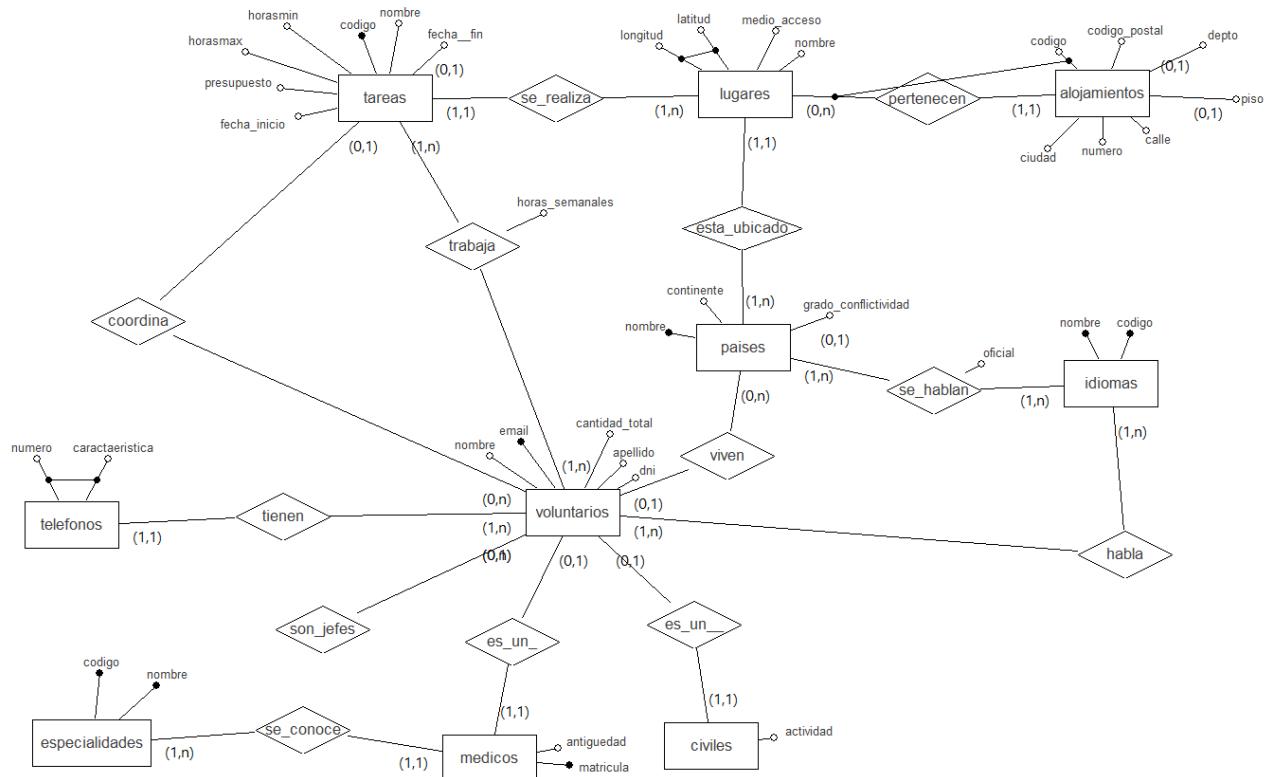
grado_conflictividad = Números enteros del intervalo [0,10]

Atributos calculados:

cantidad_total = Sumatoria de todas las horas_semanales que trabaja en las tareas cada voluntario en particular.
oficial = {S, N}

Figura 68

En la Figura 69 se encuentra el Esquema Lógico para el **Caso de Estudio de Médicos sin Fronteras**.



Dominios definidos por el usuario:

medio_acesso = {auto, 4x4, camión, avión, barco, tren}

continente = {ASIA, AMÉRICA DEL SUR, AMÉRICA CENTRAL, AMÉRICA DEL NORTE, ÁFRICA, ANTÁRTIDA, EUROPA, OCEANÍA}

grado_conflictividad = Números enteros del intervalo [0,10]

Atributos calculados:

cantidad_total = Sumatoria de todas las horas_semanales que trabaja en las tareas cada voluntario.
oficial = {S, N}

Figura 69



En la Figura 70 se encuentra el Esquema Relacional para el **Caso de Estudio de Médicos sin Fronteras**, que puede descargarse del siguiente link: <https://drive.google.com/file/d/1NL4EqnA6wJen6y6rZPM9z75T4jGY54uv/view?usp=sharing>

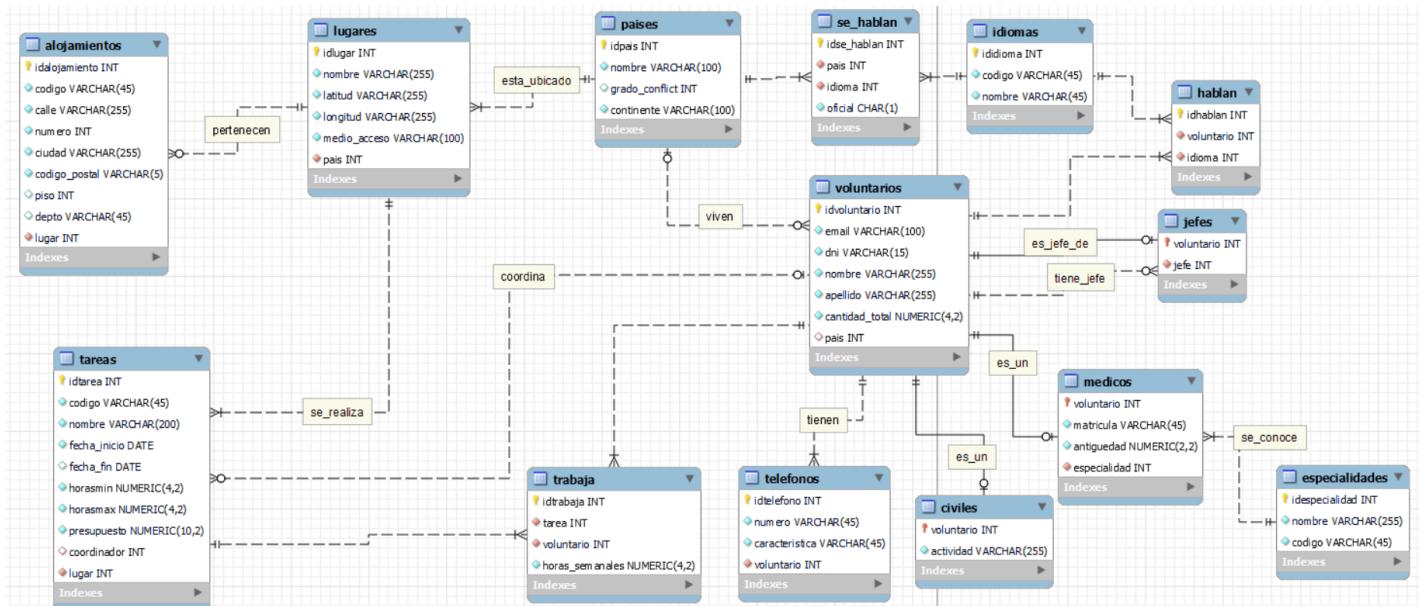


Figura 70