

Introducción

Describiremos la implementación en MySQL de los elementos del modelo relacional, a saber, tablas, atributos y relaciones, junto con las restricciones o *constraints* relacionados a las claves primarias, claves foráneas y dominios.

Nuevamente usaremos MySQL Workbench como herramienta de visualización de todos estos elementos existentes en el modelo de datos preconstruido Sakila, por lo que sugiero revisar la existencia del mismo y, si no está instalado, seguir las instrucciones ya compartidas en lecturas anteriores.

El objetivo de esta lectura es permitir un reconocimiento de la implementación de todos los componentes del modelo relacional en una base de datos concreta y apta para ser usada en una aplicación que explote estos datos.

1. Componentes del modelo relacional

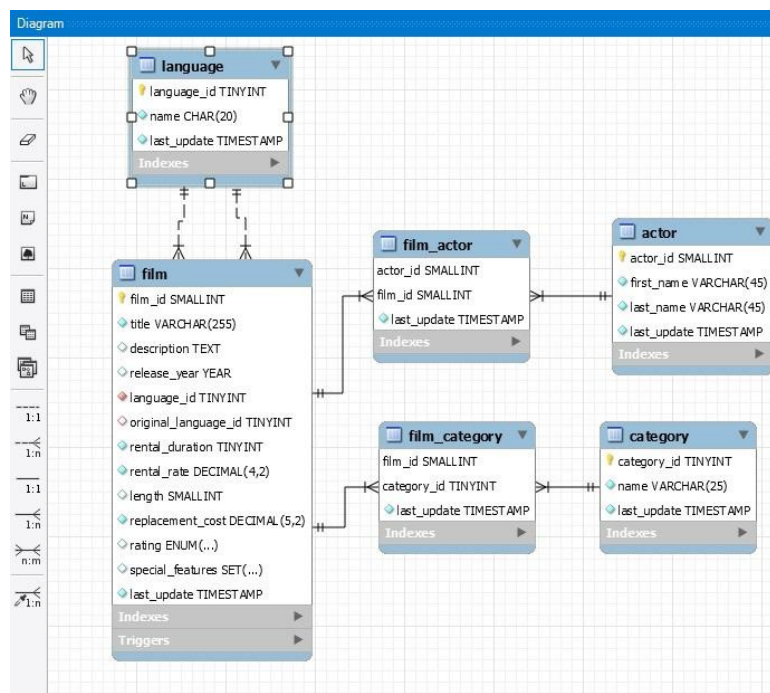
Apelando a la bibliografía obligatoria de la materia, Elmasri & Navathe (2002) y Reinosa, Maldonado, Muñoz, Damiano, & Abrutsky (2012), daremos sustento teórico describiendo brevemente distintos elementos para que sirvan como guía para entender los detalles de la implementación y las posibles diferencias respecto a lo prescripto en el modelo relacional por la necesaria abstracción de éste y la implementación en MySQL.

Encontramos en Navathe (2002) que:

El objeto básico que representa el modelo Entidad Relación es la **entidad**, que es una cosa importante del mundo real con existencia independiente. Una **entidad** puede ser un objeto con existencia física (una persona, una casa o un aula) o un objeto con existencia conceptual (una empresa, un puesto laboral o una materia). Cada **entidad** tiene propiedades específicas, llamadas **atributos** que la describen. (Elmasri & Navathe, 2002, p. 44).

En la figura siguiente, veremos a seis de las entidades ya creadas como tablas que seleccionamos de las dieciséis existentes en el esquema Sakila de la base de estudio.

Figura 1: Diagrama de tablas del esquema Sakila



Fuente: elaboración propia.

Figura 1. Diagrama de tablas. En este diagrama del modelo creado para el estudio, vemos algunas de las tablas que están creadas en el esquema de la base de datos instalada con el servidor MySQL.

En el modelo de la figura 1, también vemos los símbolos al lado del nombre de los atributos, que indican, por ejemplo, los atributos claves. Una llave amarilla, un rombo vacío, lleno o de colores indican si son obligatorios, es decir, que no pueden aceptar valores null, u opcionales, porque si pueden aceptar valores null. En el caso que tengan una restricción de clave foránea y obligatoria, el rombo es rojo. Las líneas que unen las tablas de a pares representan relaciones ya resueltas con su respectiva clave foránea, que siempre va del lado del “mucho” o del lado del “tridente” en el extremo de la relación.

Para definir con mayor precisión los componentes del modelo de la figura 1, citaremos nuestra bibliografía:

El concepto fundamental, en el Modelo Relacional, es que los datos se representan de una sola manera, en el nivel de abstracción que es visible al usuario, y es, específicamente, como una estructura tabular —conformada por filas y columnas— o como una tabla con valores.

Tabla 1: Estructura tabular

Relación Estudiantes						
Legajo	Apellido	Nombres	TipoDoc	NroDoc	IdLocalidad	Sexo
11904	González	Ana Carolina	DNI	35900342	31	Femenino
13789	López	Hugo Sebastián	DNI	29762007	27	Masculino

Fuente: Reinos, Maldonado, Muñoz, Damiano, & Abrutsky, 2012, pp. 48.

A esta estructura se la denomina formalmente relación, aunque, en lo informal y en la práctica, se la conoce con el nombre de tabla. La diferencia está en el grado de abstracción, ya que la relación cuenta con ciertas propiedades —que se verán más adelante— y la relación ya es la implementación, en la que no necesariamente se cumple con todas las propiedades de la relación, porque depende del diseño que se propone y que se implementa. Los conceptos básicos que se utilizarán, a partir de este momento, son propios del modelo y se resumen en:

- **Relación:** no es más que una representación en dos dimensiones, o de doble entrada, constituida por filas, o tuplas y columnas o atributos. Es el concepto primitivo y fundamental del modelo. La relación debe cumplir con un conjunto de restricciones o propiedades que ya han sido mencionadas. Dentro del diseño de la base de datos, las relaciones representan a las entidades que se modelaron. Las entidades poseen atributos que las distinguen y cada uno de ellos está ligado a un dominio en particular.
- **Fila o tupla:** es un hecho en la relación que contiene datos de la realidad. Por ejemplo: los datos de un alumno en particular forman una tupla de la relación “estudiantes”; los datos de un artículo pueden ser una tupla en una relación de la base de datos de una “empresa de ventas o producción”, etcétera. La tupla en una relación es una colección ordenada de elementos diferentes. Cada tupla es también una única combinación de estos elementos; por ende, las tuplas no se repiten dentro de la misma relación. Las ocurrencias de distintas instancias en la entidad se reflejan en distintas colecciones de elementos dentro de una relación y cada colección es una tupla. Podríamos decir que una instancia dentro de la entidad posee una tupla en la relación correspondiente.
- **Cuerpo:** al conjunto de tuplas de una relación se lo denomina cuerpo de la relación. El cuerpo de la relación “estudiantes” tiene dos tuplas.
- **Columna o atributo:** es una propiedad que caracteriza a cada entidad, como el color o tamaño de un artículo, el apellido o el legajo de un estudiante, etcétera. Los elementos que componen la estructura de la entidad se denominan atributos y serán irrepetibles en la entidad.
- **Cabecera:** el conjunto de atributos de una relación conforma la cabecera de dicha relación. La cabecera de la relación “estudiantes” es:

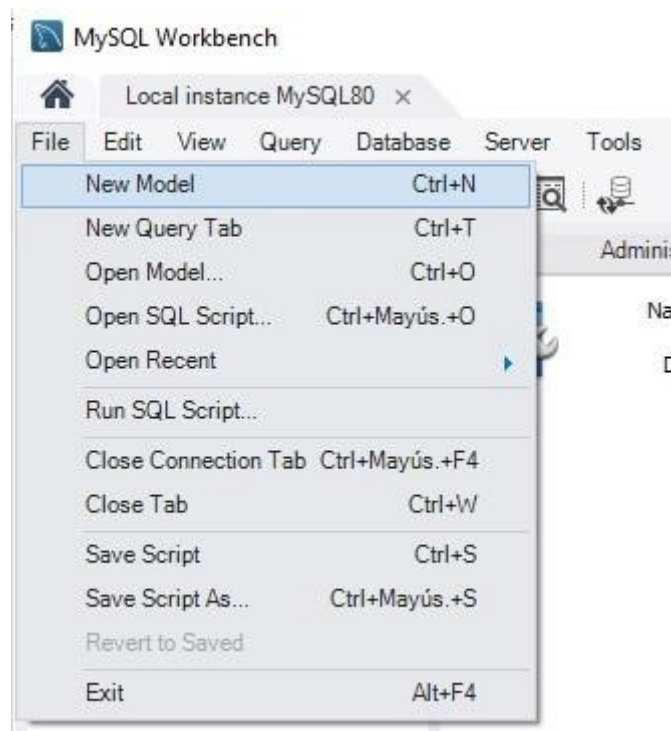
Legajo	Apellido	TipoDoc	NroDoc	IdLocalidad	Sexo
--------	----------	---------	--------	-------------	------

- Dato: es la mínima unidad que se almacena en una relación, indivisible en el concepto original del modelo almacenado en la intersección de una fila y de una columna. Por ejemplo: 35900342 es un número de documento de un alumno en la relación “estudiantes”.
- Grado: se llama así al número de columnas que conforman la relación. Es estático en el tiempo, aunque se modificará ante una necesidad de cambio dado por la realidad de la organización. En la relación "estudiantes", el grado es 7 (siete).
- Cardinalidad: así se denomina al número de tuplas o filas de una relación. Es dinámica, ya que depende del agregado o la eliminación de filas a través del tiempo. La relación “estudiantes” tiene cardinalidad 2 (dos).
- Dominio: es el conjunto de valores posibles de un atributo en la relación. Ejemplos: sexo, en la relación “estudiantes”, asume dos valores posibles (femenino, masculino); legajo se puede definir como un dato que puede asumir valores numéricos enteros positivos, sin incluir el cero (Reinosa, Maldonado, Muñoz, Damiano, & Abrutsky, 2012, pp. 48 y 49).

2. La creación de un modelo basado en tablas pre existentes

Para crear este modelo con MySQL Workbench, seguiremos los pasos ilustrados en las siguientes figuras.

Figura 2: Creación del modelo

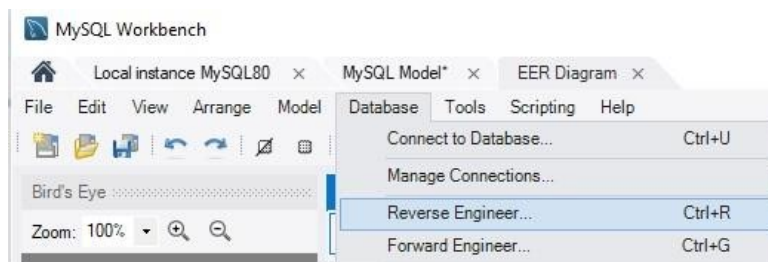


Fuente: elaboración propia.

Figura 2. Creación del modelo

Desde la opción *File* elegir *New Model*.

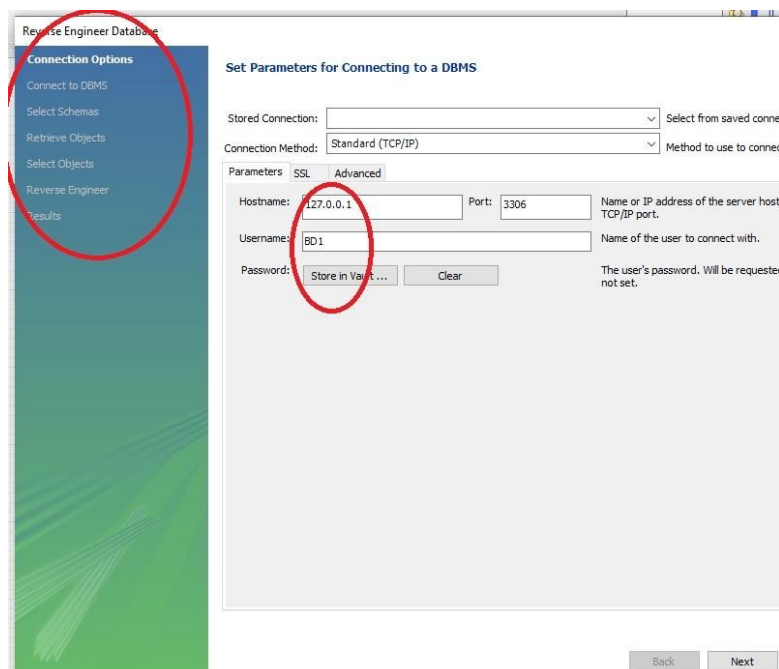
Figura 3: Elegir la acción *Reverse Engineer*



Fuente: elaboración propia.

Figura 3. Elegir la acción *Reverse Engineer*. Con esta opción, nos vamos a conectar a un esquema Sakila asociado al usuario BD1 previamente creado para que lea del diccionario de datos toda la información guardada de las tablas ya existentes.

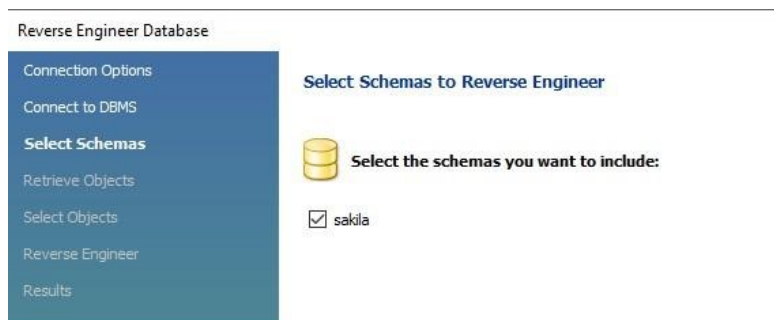
Figura 4: Conectarse como el usuario BD1



Fuente: elaboración propia.

Figura 4. Conectarse con el usuario BD1. Con esta opción nos vamos a conectar al usuario para que nos permita acceder al esquema Sakila.

Figura 5: Elegir el esquema Sakila



Fuente: elaboración propia.

Figura 5. Elegir el esquema Sakila. Con esta opción nos vamos a conectar a un esquema Sakila asociado al usuario BD1 previamente creado para que lea del diccionario de datos toda la información guardada de las tablas ya existentes.

Figura 6: Elegir las tablas del esquema seleccionado



Fuente: elaboración propia.

Figura 5. Elegir las tablas del esquema seleccionado. Con esta opción vamos a seleccionar todas las tablas para luego refinar la selección a unas pocas tablas para facilitar su lectura.

Un concepto importante en este punto: la definición de la clave primaria

En el modelo relacional es muy importante la definición de la clave primaria de cada tabla, debido a que es necesario tener una forma de especificar cómo las filas son distinguibles entre sí. Conceptualmente las entidades y sus filas no pueden ser repetidas porque son distintas y la diferencia entre ellas se debe expresar a través de sus atributos.

Por lo tanto, los valores de los atributos de una entidad deben ser tales que permitan identificar unívocamente a la entidad. En otras palabras, no se permite que ningún par de filas tenga exactamente los mismos valores de sus atributos.

Una clave permite identificar con un conjunto de atributos suficientes para lograr distinguir las filas entre sí.

Una clave compuesta es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única a una fila. Por ejemplo, el atributo id-cliente del conjunto de entidades cliente es suficiente para distinguir una entidad cliente de las otras. Así, id-cliente es una clave simple.

Es posible que conjuntos distintos de atributos puedan servir como clave, si existieran varios conjuntos así, llamaremos a todos estos claves candidatas. Supóngase que una combinación de nombre-cliente y calle-cliente es suficiente para distinguir entre las filas de la tabla cliente. Entonces, los conjuntos {id-cliente} y {nombre-cliente, calle-cliente} son claves candidatas.

Aunque los atributos id-cliente y nombre-cliente juntos puedan distinguir filas de clientes, su combinación no forma una clave candidata, ya que el atributo id-cliente por sí solo es una clave candidata.

Se usará el término clave primaria para denotar una clave candidata que es elegida por el diseñador de la base de datos como elemento principal y única, para identificar las filas dentro de una tabla. Una clave primaria es una propiedad de la tabla. Cualquier fila de una tabla no puede tener el mismo valor en sus atributos clave al mismo tiempo. La designación de una clave primaria representa una restricción o *constraint* de la tabla.

Las claves candidatas se deben designar con cuidado. Como se puede comprender, el nombre de una persona es obviamente insuficiente, ya que hay mucha gente con el mismo nombre. En Argentina, el D.N.I. puede ser una clave candidata. Como los no residentes en Argentina normalmente no tienen D.N.I., las empresas internacionales pueden generar sus propios identificadores únicos. Una alternativa es usar alguna combinación única de otros atributos como clave. La clave primaria se debería elegir de manera que sus atributos nunca, o muy raramente, cambien. Por ejemplo, el campo dirección de una persona no debería formar parte de una clave primaria, porque probablemente cambiará. Los números de D.N.I., por otra parte, es seguro que no cambiarán. Los identificadores únicos generados por empresas generalmente no cambian, excepto si se fusionan dos empresas; en tal caso el mismo identificador puede haber sido emitido por ambas empresas y es necesaria la reasignación de identificadores para asegurarse de que sean únicos.

Un dominio es el conjunto de valores que puede tomar un atributo.

Verdadero.

Falso.

Referencias

Elmasri, R., & Navathe, S. (2002). *Fundamentos de Sistemas de Bases de datos*. Madrid: Pearson Educación.

Reinosa, J., Maldonado, C., Muñoz, R., Damiano, L., & Abrutsky, M. (2012). *Bases de datos*. Buenos Aires: Alfaomega Grupo Editor Argentino.