×

CODING

Python Full Stack (b2u-4-medio-principe-de-gales-2025)

Fundamentos de Python 5/46 > Flask 2/21 > MySQL 1/46 ^

Instalaciones / (Mac) Instalación MySQL

Instalaciones / (Windows) Instalación MySQL

Instalaciones / Comprobando MySQL Workbench

ERDs / Descripción General

ERDs / Diseño de Bases de datos

ERDs / Relaciones uno a uno

ERDs / Me gusta

Contenido del curso

ERDs / Creación de ERDs en MySQL

Workbench ERDs / Convenciones

ERDs / Tipos de datos

ERDs / Usuarios (Práctica)

ERDs / Relaciones uno a muchos y muchos a

ERDs / Relaciones de muchos a muchos

ERDs / Normalización

ERDs / QUIZ 1

ERDs / Blog Simple

ERDs / Reseñas de restaurantes

ERDs / Productos

ERDs / Insignias

ERDs / Estudiantes y Cursos (Core)

ERDs / Canciones (Core)

ERDs / Seguidores (Práctica)

(Opcional)

ERDs / Administrador de Proyectos

ERDS / Blogs (Opcional)

ERDs / Administrador de Usuarios (Opcional)

ERDs / Normalización (Opcional)

MySQL Queries / SQL MySQL Queries / Conectarse al servidor

MySQL Queries / Importar estructura y/o

MySQL Queries / Ingeniería Avanzada

MySQL Queries / SELECT

MySQL Queries / INSERT

MySQL Queries / UPDATE

MySQL Queries / DELETE MySQL Queries / Funciones

MySQL Queries / JOINS

MySQL Queries / LEFT JOIN

MySQL Queries / Restricciones de las llaves foráneas

MySQL Queries / Familiarizándonos con

MySQL Queries / Exportar

MySQL Workbench (Práctica)

MySQL Queries / Usuarios (Práctica)

MySQL Queries / Estudiantes y Cursos (Core)

MySQL Queries / Canciones (Práctica)

MySQL Queries / Seguidores (Práctica) MySQL Queries / Países MySQL (Core)

MySQL Queries / Sakila (Opcional)

Flask + MySQL 0/13 > AJAX

Deployment 0/18 >

0/1 > Examen de Certificación

## Relaciones de muchos a muchos

## Objetivos

Aprender qué son las relaciones muchos a muchos y cuándo utilizarlas

Continuemos con el esquema en el que hemos trabajado; por lo pronto tenemos una tabla dedicada a la información de cada pedido, sin embargo aún nos hace falta dar seguimiento a los productos relacionados al pedido.

Para esto crearemos una tabla de productos que incluye el nombre y descripción de cada producto que el usuario puede agregar a su pedido.

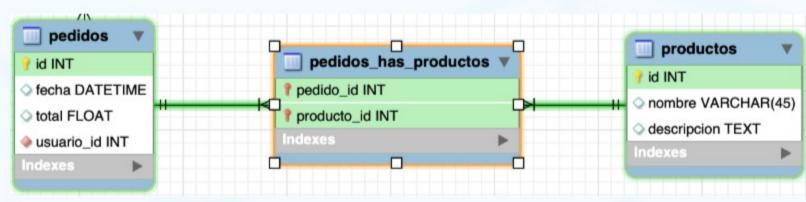
productos		
id	nombre	descripción
222	lapiz	lapiz que te ayudará a escribir ERDs
223	libreta	para escribir todas tus notas de MySQL
224	clip	sostiene tus hojas
225	bolígrafo	tus notas no se borraran con nada
226	marcatextos	para subrayar lo más importante
227	tijeras	recorta todo lo que necesites

¿Qué tipo de relación tenemos aquí entre pedidos y productos? Analicemos en voz alta: un pedido puede incluir muchos productos (el pedido 551 puede incluir el lápiz, el clip, las tijeras) y un producto puede ser parte de muchos pedidos (el clip puede ser parte del pedido 551, 553 y 554). Por esto, la relación entre pedidos y productos es de muchos a muchos.

En las relaciones de **muchos**, utilizamos una tabla conectora (también conocida como tabla de unión). La tabla incluye pedido\_id y producto\_id, que hacen referencia a los identificadores de las tablas de pedidos y productos; de esta forma podemos determinar los elementos que se unen.

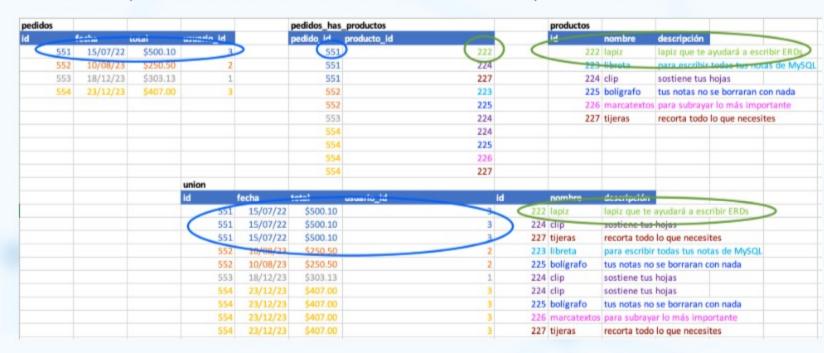
pedidos_has_productos			
pedido_id	producto_id		
551	222		
551	224		
551	227		
552	223		
552	225		
553	224		
554	224		
554	225		
554	226		
554	227		

Representamos este tipo de relaciones de la siguiente forma:



## La capacidad de SQL

De la misma manera en la que hemos unido los dos tipos de relaciones pasadas, aquí podremos hacerlo también. Las llaves foráneas que encontramos en la tabla de unión serán la manera en la que uniremos las dos tablas.



## Ejemplos de relaciones muchos a muchos

■ Usuarios y pasatiempos: un usuario puede tener muchos pasatiempos, un pasatiempo puede pertenecer a muchos usuarios.

Actores y películas: Un actor puede actuar en muchas películas, una película puede tener muchos actores.

← Anterior

Siguiente →



