

PRÁCTICA SQL Y DW

1. DIAGRAMA ENTIDAD RELACIÓN

Realizar el diagrama entidad relación con el que poder modelar una base de datos para KeepCoding, es decir, recogiendo datos de alumnos, bootcamps, módulos, profesores... Se debe entregar un pdf con el diagrama y una pequeña explicación de este.

Hola Alex,

Para presentarte mi solución para este primer ejercicio he decidido dividir la explicación en los grandes subconjuntos de la base de datos que he creado para que pueda explicarte con detalle cada uno de ellos. Estos subconjuntos son: usuarios y roles, credenciales, domicilio, bootcamps y asignaturas. Adjunto a continuación una imagen que muestra la base de datos completa y una tabla que explica la función de cada uno de ellos.

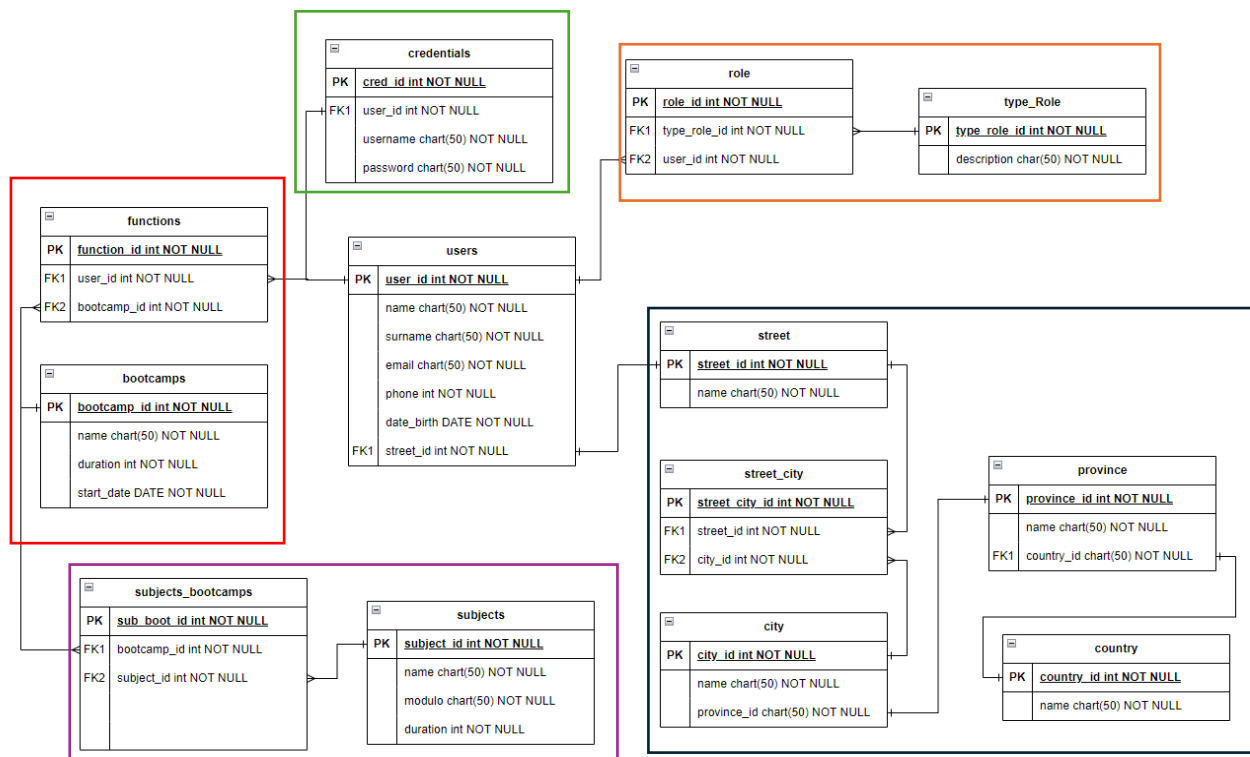


Imagen 1. Base de datos de KeepCoding y sus subconjuntos.

Color	Subconjunto	Explicación
Naranja	Roles	Este subconjunto se encarga de gestionar los roles de los usuarios.
Verde	Credenciales	Subconjunto creado por una única tabla en el que se almacenan los username y contraseñas de cada usuario.
Rojo	Bootcamps	Subconjunto que almacena toda la información de los bootcamps de KeepCoding.
Morado	Asignaturas	Este subconjunto almacena la información acerca de todas las asignaturas que componen los bootcamps.
Azul oscuro	Domicilio	Subconjunto encargado de gestionar los domicilios de los usuarios.

USUARIOS

Lo primero que decidí fue crear los usuarios que componían la academia de KeepCoding. Para ello, al principio creé tres tablas distintas: una para alumnos, otra para profesores y otras para el personal de KeepCoding que trabajan en la academia que no son profesores, como pueden ser RRHH, contabilidad, marketing... Este tercer grupo lo denomine el grupo de administración.

Al empezar a crear las tres tablas me percaté de que los campos que estaba creando para cada tabla eran prácticamente los mismos (nombre, apellido, correo, teléfono...). Es por eso por lo que para evitar la creación de tablas que eran similares, deseché esta idea de tener distintas tablas para las personas y decidí crear una única tabla para los usuarios. Esta tabla es una tabla que contiene la información personal de los usuarios. Adjunto seguidamente una imagen de esta para que puedas observar los atributos que le he creado.

Users	
PK	<u>user_id int NOT NULL</u>
	name char(50) NOT NULL
	surname char(50) NOT NULL
	email char(50) NOT NULL
	phone int NOT NULL
	date_birth DATE NOT NULL
FK1	street_id int NOT NULL

Desde mi punto de vista, el hecho de tener una única tabla centralizada con toda la información de los usuarios es mejor porque en primer lugar me parece que es más eficiente que tener distintas tablas que contienen más o menos la información y por otro lado porque creo que esto permite que la estructura de la base de datos sea más sencilla, lo cual puede suponer una ventaja en caso de que a futuro se desee cambiar algo de ella. Por ello, esta tabla es la tabla “principal” de la base de datos ya que hace de unión entre los distintos subconjuntos anteriormente mencionados.

Imagen 2. Tabla users

ROLES

Agrupar todos los usuarios en una misma tabla posee la ventaja de que, desde mi punto de vista, es más eficiente y más “limpio” pero posee la desventaja de que sea un poco más confuso ya que en una misma tabla se agrupan todos los usuarios sin importar su rol en la academia. Es por eso por lo que para solucionar esta desventaja lo que planteé fue crear dos tablas: una que recoge los roles que existen en la academia (*type_role*) y otra tabla intermedia que especifica los roles de cada usuario (*roles*).

La tabla *type_role* es muy simple ya que simplemente posee dos campos: el número identificador del rol y su descripción asociada. A priori, había planteado crear seis roles: estudiantes, profesores, administración, futuros estudiantes, futuros profesores y futuros administración. Como he mencionado anteriormente, administración engloba a todas las personas que trabajan en KeepCoding que no son ni profesores ni estudiantes como pueden ser RRHH, la gente encargada de los procesos de selección, etc.

Por otro lado, la tabla *role* recoge los roles asociados a cada usuario. La solución la he planteado de tal modo que la relación entre los usuarios y los roles sea de relación N:M ya que puede darse el caso de que una persona pueda tener más de un rol asociado. Por ejemplo, un profesor de un bootcamp puede ser a su vez estudiante de otro bootcamp o una persona que trabaja en administración también puede ser a su vez estudiante. Sé que esta situación es remota, pero debido a que era posible crear esta solución, me parecía conveniente crear así la solución para que fuera más robusta. De este modo, y siguiendo con el ejemplo anterior, el usuario 1 podría tener asociados los roles 1 (estudiante) y el 3 (administración).

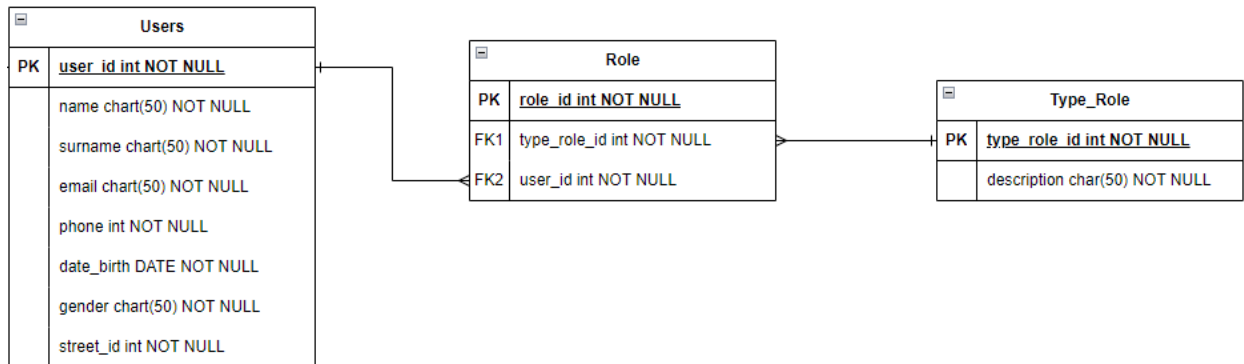


Imagen 3. Relación entre usuarios y roles.

CREDENCIALES

Cada persona que trabaja o estudia en KeepCoding, posee unas credenciales personales para poder acceder a su espacio personal. Por ello, he creado una tabla que recoger el ID del usuario, su *username* asociado y su contraseña. Es una tabla sencilla y la relación con la tabla de usuarios es de 1:1 ya que no puede ser que un usuario posea dos cuentas en KeepCoding.

En sí, podía haber creado un campo en la tabla de usuarios que contuviera el ID de la credencial y así no tener que crear el campo *user_id* en la tabla de credenciales, pero desde mi punto de vista, creo que es mejor tener en una misma tabla el *user_id*, con los *username* y las contraseñas para que en caso de que haga falta cambiar alguna credencial, no sea necesario hacer un *join* con la tabla *users*.

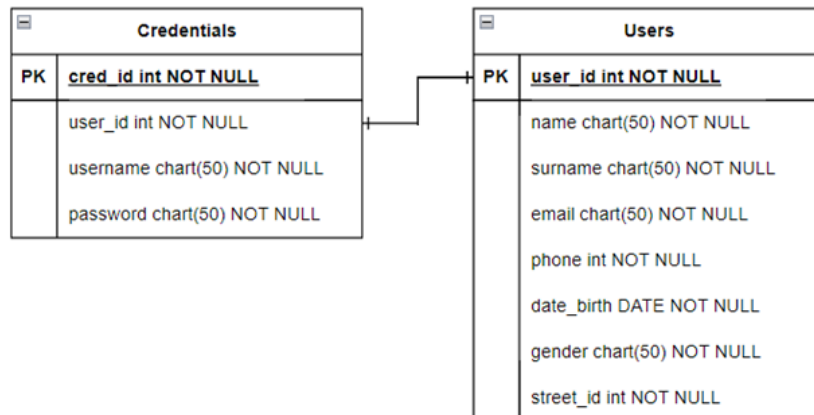


Imagen 4. Relación entre usuarios y credenciales.

DOMICILIO

Por último, y para completar los usuarios, decidí crear un campo con la calle del usuario y de ahí como vimos en clase, crear tablas para conocer la ciudad, el estado o comunidad autónoma y el país. Me parecía interesante crear esta tabla ya que creo que es importante conocer la ubicación de la gente que trabaja en KeepCoding ya sea por la diferencia horaria, para conceder becas o por el motivo que pueda ser.

Como puede apreciarse en la imagen todas las relaciones son 1:1 excepto por la relación calle-ciudad que es 1: N porque puede darse el caso de que una misma calle este en distintas ciudades. Por ejemplo, el nombre calle mayor puede estar en San Sebastián o Madrid. Por eso, creé la tabla *street_city* la cual permite que se dé esta relación.

Por los demás la relación 1:1 porque entiendo que dos ciudades puedan estar en la misma provincia. Puede darse el caso de que haya dos ciudades que se llamen igual en el mundo (como Valladolid, por ejemplo) pero no puede ser que haya dos ciudades con el mismo nombre en la misma provincia/estado/condado (he denominado la tabla como provincia por incluir un nombre genérico). Es por eso por lo que la relación entre ciudad y provincia es 1:1.

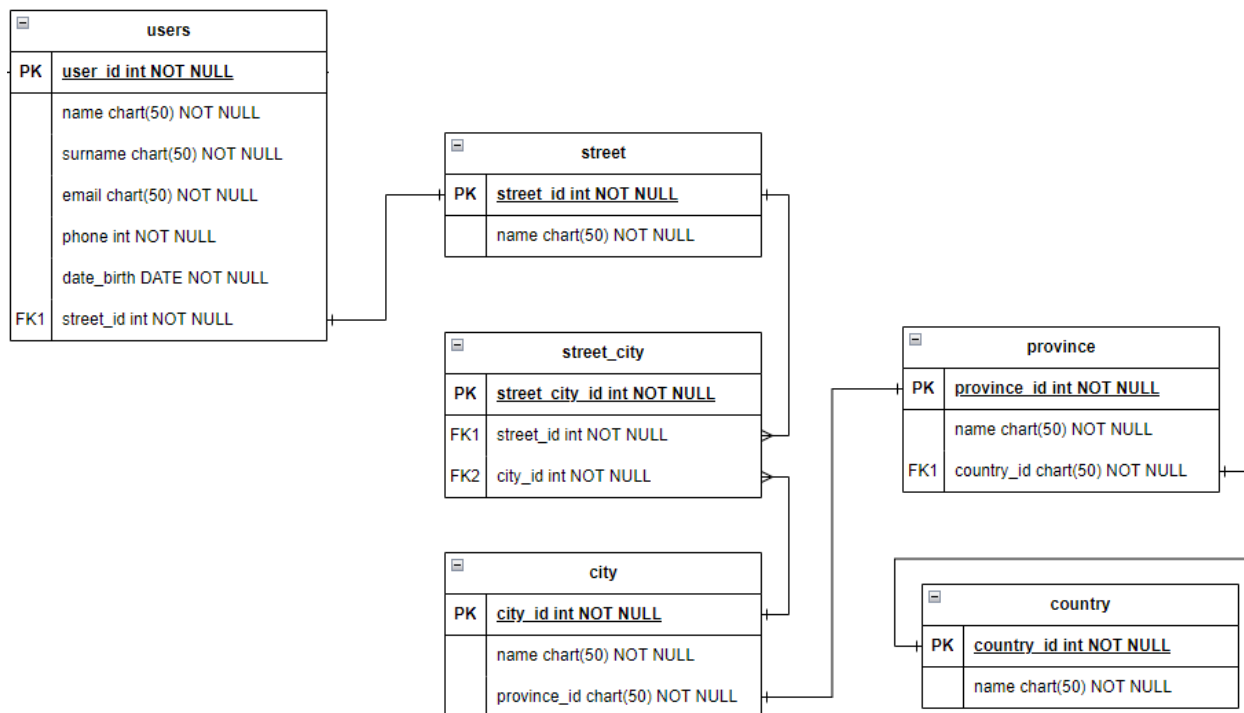


Imagen 5. Relación entre usuarios y domicilio.

BOOTCAMPS

Una vez definidos los usuarios con todos sus atributos y con sus credenciales y ubicaciones, la siguiente entidad que consideré que era necesaria crear era la entidad Bootcamp. Al final el valor KeepCoding reside en sus bootcamps y por ello consideré que era buena idea crear una tabla que contuviera la información de los bootcamps. Los atributos que he creado para esta tabla son los atributos que aparecen en el folleto de información cuando una persona solicita información sobre un bootcamp.

La tabla bootcamp a su vez está relacionada con la tabla de usuarios ya que:

- Si el usuario es estudiante, es necesario saber el bootcamp que estudia.
- Si el usuario es profesor, es necesario saber el bootcamp que imparte.
- Si el usuario es de administración, es necesario conocer el bootcamp que administra. Por ejemplo, igual una persona de RRHH se encarga de solamente dos bootcamps.
- Si el usuario es futuro estudiante, es necesario saber el bootcamp que estudiará.
- Si el usuario es futuro profesor, es necesario saber el bootcamp que impartirá.
- Si el usuario es de administración, es necesario conocer el bootcamp que administrará.

Además, como he comentado anteriormente, puede darse el caso de que una persona estudie un bootcamp y que a su vez imparta un bootcamp y por lo tanto la relación entre ambas tablas es N: M. Por ello para realizar la unión entre ambas tablas, decidí crear una tabla intermedia parecida a la tabla *role*, solo que en este caso denominada como *function*, en donde se pueda especificar los usuarios y los roles que tiene asociados.

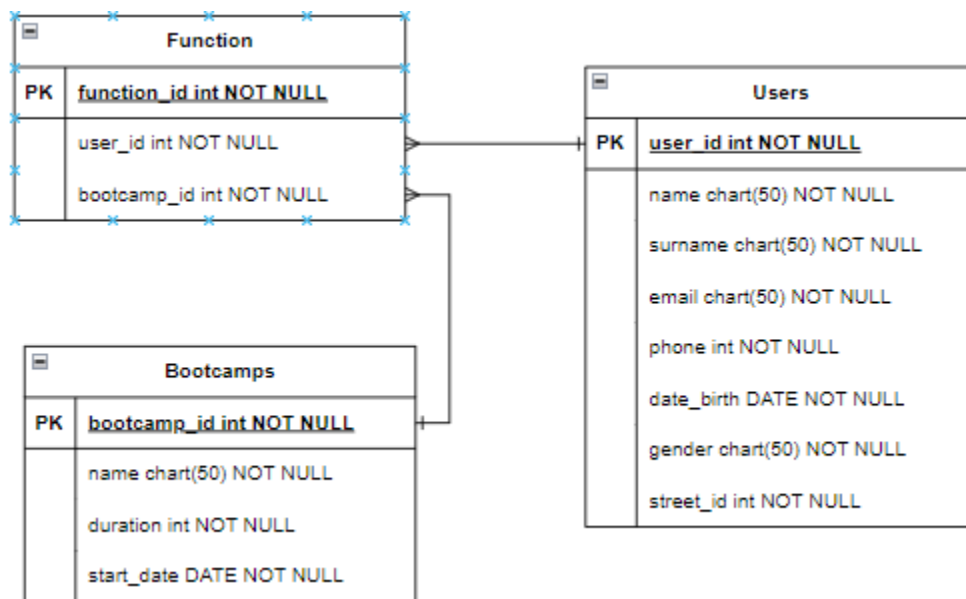


Imagen 6. Relación entre usuarios y bootcamps.

ASIGNATURAS

Por último, los bootcamps a su vez se componen de asignaturas y cada asignatura puede pertenecer a varios bootcamps. Por ejemplo, esta asignatura de SQL pertenece tanto al bootcamp de bases de datos como al bootcamp de inteligencia artificial y, por lo tanto, se trata otra vez de una relación N: M. Al igual que antes, he creado una tabla intermedia entre la tabla bootcamps y la tabla asignatura para que pueda darse esta unión.

Quiero destacar que los atributos de la tabla de asignaturas los he elegido en base al folleto de información que nos enviaron cuando formalizamos la matrícula.

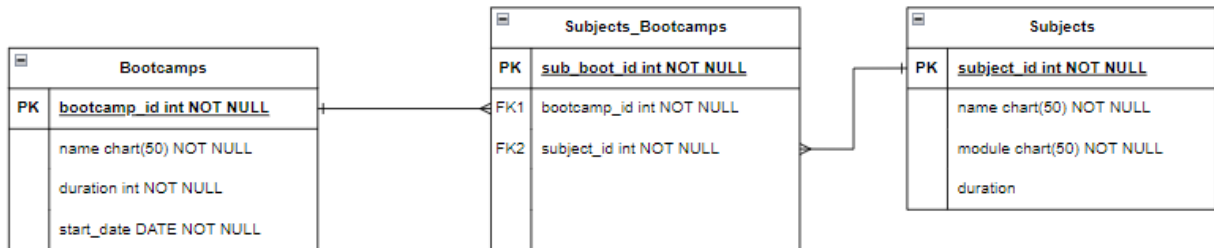


Imagen 7. Relación entre asignaturas y bootcamps.

RESUMEN

Adjunto a continuación una imagen en la que se pueda observar la base de datos al completo, con todas sus relaciones entre tablas y las claves primarias y foráneas de cada tabla.

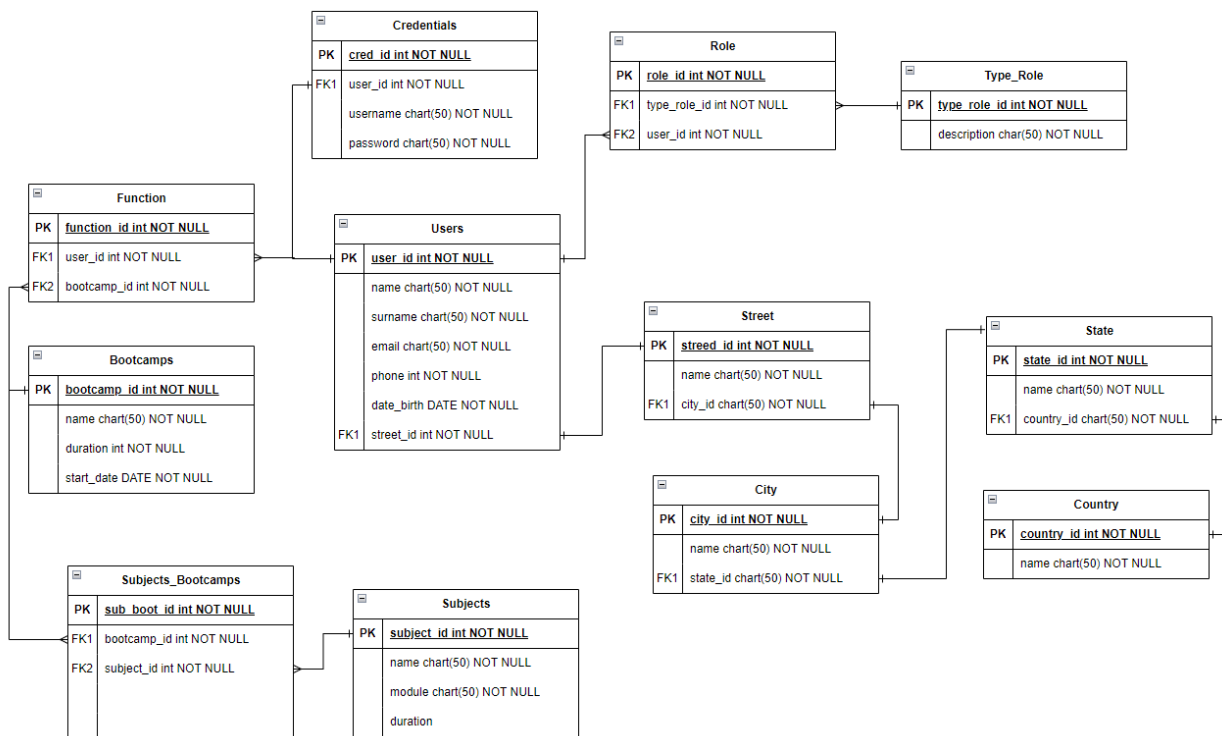


Imagen 8. Base de datos de KeepCoding.