

Diagrama Entidad-Relación

PRÁCTICA SQL

Ruben Cerezo Cuesta

En este diagrama entidad-relación vemos la base de datos de bootcamps dentro de la empresa Keepcoding. La utilidad del diagrama es poder entender de un solo vistazo las relaciones entre todos los elementos que interactúan dentro de esta empresa. Para esto, hay que entender que cada recuadro representa una **tabla** de la base de datos, y cada línea dentro de este, un **campo** de dicha tabla

Como hemos estudiado durante el curso, una **base de datos (BBDD)** en cada tabla debe tener una **Primary Key**, y si interactúa con otras tablas, la manera de incluirlas es a través de incluir un campo donde esté la **Foreign Key**. Esto lo vemos en el diagrama representado a través de las siglas **PK** y **FK**, además de que vemos que la tabla donde está la FK es donde apunta la flecha que las relaciona

Las tablas y filas que representamos en esta tabla son los siguientes:

1. Bootcamp

- **PK:** bootcamp_id
 - Atributos: nombre_bootcamp, Edición, fecha_inicio, fecha_fin
 - Gracias al campo Edición, seremos capaces de agrupar en una misma consulta a todos los alumnos de una misma edición
 - Relación:
 - Dentro de cada registro de la tabla alumnos, incluiremos la FK de su bootcamp, de manera que podremos conseguir los datos del alumno y qué está estudiando
-

2. Alumnos

- **PK:** alumno_id
- Atributos: nombre_alumno, apellidos_alumno, facturación (mes/año), email_alumno, telefono_alumno, discord_alumno, Curso
 - Puede resultar extraño incluir en esta tabla el campo discord_alumno, pero considero que es útil dado que el email y discord son los medios de comunicación principales de keepcoding
- **FK:** bootcamp_id
 - Gracias a esta FK, podemos comprobar el bootcamp al que pertenece el alumno, y así relacionarlo con sus asignaturas y notas
- **FK:** pais_id (está asociado a un país)
 - He decidido incluir una relación a la tabla país, dado que en Keepcoding vemos alumnos de distintas nacionalidades, y el huso horario en el que viven puede afectar a la comunicación (por ejemplo, para evitar llamarle a horas en las que en su país sea de noche)
- **FK:** notas:

- Las notas del alumno se relacionan con este, y se representan con una flag (is_passed) que puede ser True o False, asociando una nota a cada asignatura y alumno

3. Profesores

- **PK:** profesor_id
- Atributos: nombre, apellidos, sueldo
- **FK:** asignaturas_id
 - Dado que cada profesor enseña una asignatura, estos dos campos están relacionados entre sí. Ya que una BDD permite asociar más de un profesor, no hay problema en el caso de asignaturas con más de un profesor, como Matemáticas 101, ni tampoco en asociar asignaturas a más de un bootcamp, como es el caso de SQL avanzado, donde hay alumnos de distintos bootcamps

4. Asignaturas

- **PK:** asignaturas_id
- Atributos: nombre, Transversal (S/N), Mes (mes), Horas lectivas
- **FK:** bootcamp_id (a qué bootcamp pertenece)
 - Con esta relación podemos ver a qué bootcamp pertenece cada asignatura
- **FK:** profesor_id (quién la imparte)

5. Notas

- **PK:** notas_id
 - Esta es una tabla intermedia entre los campos asignaturas y alumnos que nos permite ver si el alumno ha aprobado o no
- Atributos: is_aprobado (True/False)
- **FK:** asignaturas_id (notas de qué asignatura son)

6. País

Gracias al huso horario, podemos comprobar si queremos llamar al alumno o trabajador en determinado horario

- **PK:** pais_id
- Atributos: País, Huso horario
- Relación:

- Asociado a **alumnos**
 - Asociado a **personal no docente**
-

6. Personal no docente

- **PK:** trabajador_id
- **Atributos:** Nombre_trabajador, apellidos_trabajador, email_trabajador
- **FK:** país_id

En esta tabla, agregamos a todos los trabajadores cuyo rol no sea docente (administrativos, comerciales, personal de mantenimiento, etc)

