

Modelado & SQL fundamentos



KEEPCODING

Tech School



¿Qué vamos a ver?

Día 1

1. Fundamentos de diseño entidades, atributos etc.
2. Concetos de Diagrama entidad-relación
3. Normalización
4. Ejercicios prácticos

¿Qué vamos a ver?

Dia 2. Fundamentos de SQL

1. DDL (create, drop etc)
2. DML (Select , Update. Delete , Insert etc)
3. Ejercicios



¿Qué vamos a ver?

1. Día 3.

1. Joins entre tablas
2. Ejercicio práctico: Normalización de la base de datos empleados.
3. Leer entre todos la practica a entregar

Modelado de datos



¿Qué es una Base de datos?

Una base de datos es una recopilación organizada de información (datos estructurados) que se almacena de forma electrónica.

Una base de datos es gestionada por un “Sistema de gestión de base de datos” (DBMS o SGDB).

Un sistema de base de datos es una mezcla del DBMS + estructuras de datos + los datos.



¿Qué difiere una Hoja Excel y una BBDD?

Las hojas de cálculo se crearon principalmente para que sea gestionada por un usuario o un grupo muy pequeño.

Las BBDD permiten tener muchos usuarios accediendo y consultando los datos de forma rápida y segura al mismo tiempo

Diferencias:

- Como se almacenan y manipulan los datos
- Quien puede acceder a los datos
- Cuantos datos puede acceder. Las BBDD no tienen limite.

¿Qué es un DBMS o SGBD?

Un sistema gestor de base de datos es un software (libre o de pago) que permite crear y administrar las base de datos.

- Permite crear estructuras de datos
- Permite manipular los datos
- Seguridad de acceso a los datos
- Integridad en las operaciones sobre estructura y datos

DBMS más comunes



ORACLE



¿Qué es el Lenguaje SQL?

SQL = Lenguaje de consulta estructurada

El SQL es un lenguaje de programación que utilizan las bases de datos relaciones para manipular y definir las estructuras de datos, así como para la gestión de los datos.

SQL fue creado por IBM en la década de 1970 y se sigue usando mucho a día de hoy.



Entonces... ¿ahora nos ponemos con SQL?

La respuesta es **NO**



Antes debemos aprender a modelar datos. Modelar datos consiste en crear una base de datos que de respuesta a un problema o requisito.

Ejemplo: Si quiero almacenar los coches de todos mis vecinos. Primero tendremos que analizar que información necesitamos almacenar para tener todos los datos. Este proceso de análisis se denomina Modelización o Diseño de base de datos.



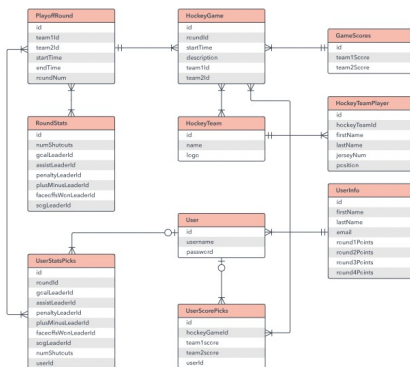
DISEÑO DE BASE DE DATOS

¿Qué es el diseño de una base de datos?

- Consiste en dar una solución a un problema concreto, donde podamos almacenar la información requerida por el problema.
- Para dar solución vamos a usar el Modelo Entidad-Relación (E-R)

¿Qué es el Modelo Entidad Relación?

- Es un diagrama que ilustra las diferentes entidades y como se relacionan entre si.
- Son un reflejo de la estructura gramatical del problema y se suele emplear las entidades (sustantivos) y las relaciones como verbos.



Usos del diagrama

- Para dar soluciones a problemas de base de datos. Se usa para encontrar analizando el diagrama problemas de lógica.
- Análisis de necesidades y procesos de negocio. Cualquier proceso de negocio que use datos se aprovecha de los diagramas para simplificar los procesos y mejorar los resultados.
- Reingeniería de procesos de negocio. Ayudan a analizar las base de datos empleadas y usarlo como base para configurar una nueva base de datos
- Investigación para análisis de datos en procesos de investigación



Diagrama ER

Un diagrama ER está compuesto por:

- Entidades
- Atributos
- Relaciones y cardinalidad

Vamos a usar : <https://app.diagrams.net>



Diagrama ER: Entidades

Una entidad se puede definir como una persona, objeto, concepto u evento, que pueda ser almacenado.

Ejemplo: Un cliente, estudiante, usuario, coche, producto

Diagrama ER: Enunciado problema

Enunciado:

Queremos gestionar los coches de empresa que se puede dar a un empleado. A un empleado solo se le puede dar un coche de empresa.

Se quiere poder:

- Saber los Kms que tiene el coche, año de compra, marca del coche, modelo, matricula.
- Saber el empleado que tiene asignado un coche, fecha de asignación

Diagrama ER: Entidades

Queremos gestionar los coches de empresa que se puede dar a un empleado. A un **empleado** solo se le puede asignar un coche de empresa.

Se quiere poder:

- Saber los Km que tiene, año de compra, marca del **coche**, matricula.
- Saber el empleado que tiene asignado un coche, fecha de asignación

Diagrama ER: Relaciones

Como las entidades interactúan o asocian entre si. Se suelen usar los Verbos.

Del Enunciado:

Queremos gestionar los coches de empresa que se puede dar a un empleado. A un empleado solo se le puede **asignar** un coche de empresa.

Se quiere poder:

- Saber los Km que tiene, año de compra, marca del coche, matricula.
- Saber el empleado que tiene asignado un coche, fecha de asignación

Diagrama ER: Atributos

- Son las propiedades o características de una entidad.
- En Excel son las columnas que creamos para añadir datos.
- Tipos de atributos:
 - Simples : por ejemplo el nombre, apellidos. Son valores que no pueden divisirse
 - Compuesto: Puede descomponerse en otros mas sencillos.
 - Múltiples: Puede tener mas de un valor. Por ejemplo un cliente puede tener N telé. (solo 1).
 - Opcionales: Pueden tomar o no valor. Es decir, si pueden o no quedar vacíos
 - Obligatorios: Deben tomar siempre valor

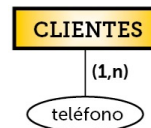
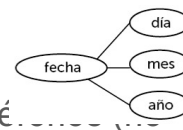


Diagrama ER: Clave Primaria

- Se trata de uno o más atributos de una entidad cuyos valores son únicos. Los elementos que haya en esta entidad tienen ese/esos atributos clave primaria (PK) un valor diferente.
- Si se intenta repetir un elemento con una clave primaria existente se produce un error por parte del SGBD
- Son atributos fundamentales y necesarios
- Todas las entidades tienen una PK (de un elemento o varios)
- Hay que elegir bien la PK de cada entidad
- Los valores de la PK no pueden estar a NULL
- Los valores de la PK no se pueden cambiar
- La PK tiene que tener valores cuando se inserta el dato

Diagrama ER: Cardinalidad

- Define los atributos numéricos de la relación entre dos entidades
- Los tipos de cardinalidad mas comunes son:
 - 1 a 1. Relación Un dato con un dato. Ejemplo un estudiando que tiene solo una dirección de email (no puede tener más)
 - 1 a N. Relación de 1 a muchos, seria un estudiante que tiene mas de una dirección de mail o por ejemplo tienen muchos cursos. Es necesario pasar de atributo a entidad nueva.
 - N a N. Relación de muchos a muchos. Por ejemplo si tenemos la entidad estudiante y asignaturas, tendremos que tener una entidad que relaciones Estudiante y asignaturas.

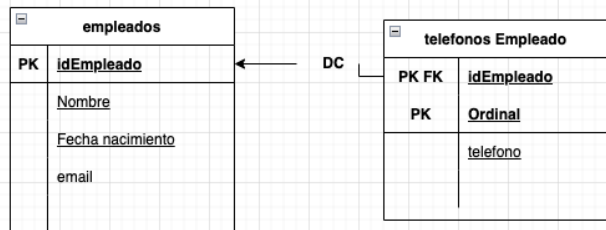
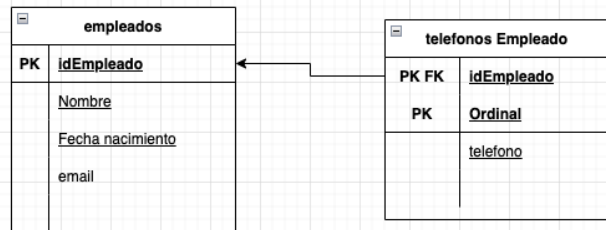
Diagrama ER: Claves Foráneas (FK)

La clave foránea o FK identifica una columna o grupo de columnas de una tabla con las de otra tabla. Establece una relación de integridad referencial.

Las columnas de la tabla referenciada o hija, debe ser PK.

Si se intenta añadir un dato en un atributo que es FK que no existe en la tabla referenciada, no se permite. (integridad referencial = integridad de los datos).

Diagrama ER: Eliminación en cascada (DC)



La integridad referencial por defecto no me va a dejar eliminar un empleado si tiene datos en la tabla de teléfonos. Esto lo hace el SGBD

Si se desea que al eliminar un empleado se borre automáticamente los datos de la tabla de teléfonos, hay que indicar al SGBD que queremos Delete Cascade.

Integridad referencial y datos

Con las PK, Claves Foráneas (FK), Cardinalidad de las relaciones, Eliminación en cascada (DC) etc. conseguimos lo que se llamada integridad referencial.

Integridad referencial:

- No podemos añadir códigos en las FK inventados.
- No se puede borrar un datos si se usa en un FK. Esto me permite que no deje de existir un código referencia. Ejemplo si borramos genero de las personas.
- En casos concretos “funcionales” se requiere Delete Cascade, pero lo tengo que indicar explícitamente.

Ejercicio 1 – Diseñar modelo de datos

Somos KCFLix una plataforma de video stream de series. Necesitamos poder tener las series (nombre, año creación), las temporadas de una serie y los actores que hay en una serie. De los actores necesitamos tener el nombre, sexo , url de su web y url de su foto.

Tener en cuenta:

- Un actor puede estar en más de una serie
- Un actor puede estar en una temporada y en otra temporada ya no estar

Hacerlo en papel y tenéis 5 minutos

Ejercicio 1 - Solución

Somos KCFLix una plataforma de video stream de **series**. Necesitamos poder tener las series (nombre, año creación), las **temporadas** de una serie y los **actores** que hay en una serie. De los actores necesitamos tener el nombre, sexo , url de su web y url de su foto.

Tener en cuenta:

- Un actor puede estar en más de una serie
- Un **actor puede estar en una temporada** y en otra temporada ya no estar

Hacerlo en papel y tenéis 5 minutos



NORMALIZACIÓN

Normalización

- Normalizar las bases de datos relacionales es un tema sumamente importante
- Las formas normales (FN) son mecanismos que proporcionan una serie de criterios para encontrar inconsistencias en nuestro modelo de datos.
- Existen 4 formas normales principales.
- Normalizar es descomponer.

Primera Forma Normal: 1FN

- Nos ayuda a eliminar los valores repetidos y no atómicos dentro de la base de datos
- Una tabla esta en 1Fn si:
 - Los atributos son atómicos. Un atributo es atómico si los elementos son simples e indivisibles
 - Debe tener una clave primaria para que no haya registros repetidos.
 - No hay orden de arriba abajo en las filas (registros como están introducidos)
 - Un atributo (o campo) no puede tener varios valores. No se pueden meter N teléfonos. En este caso se resuelve mediante una tabla nueva relacionada de teléfonos.

Segunda Forma Normal: 2FN

- La tabla debe estar en 1FN
- Datos en una misma tabla que pertenecen a diversas entidades, cada entidad tiene un campo clave separado.

Tercera Forma Normal: 3FN

- La tabla debe estar en 2FN
- No tiene ninguna dependencia funcional transitiva en los atributos que no son PK.
- Esto quiere decir que aquellos datos que no pertenecen a la entidad deben tener independencia de los demás teniendo su propio campo clave.

Cuarta Forma Normal: 4FN

- La tabla debe estar en 3FN
- Los campos de multiple valores se identifican con clave única
- Este FN trata de eliminar registros duplicados en la entidad, teniendo la entidad datos únicos, y se realiza mediante el uso de claves foráneas.

Preguntas de lo visto

- ¿Esta ese modelo totalmente normalizado como hemos hecho?
 - NO. Se puede mejorar .
 - Nivel de un curso, se puede sacar como entidad propia de niveles y relacionar mediante FK
- ¿Puedes ver limitaciones en el modelo final?
 - Un alumno solo puede hacer 1 curso. ¿Y si quiero hacer varios cursos?
 - Te meten 2 veces como alumnos, pero estaríamos duplicando información
 - Habría que modificar el modelo para que lo permita
 - ¿No estaría mejor definir un curso las materias que tiene?
 - Depende del requisito funcional

Ejercicio 2 – Diseñar modelo de datos

Queremos gestionar los alumnos de Keepcoding. Para ello queremos tener los alumnos, los bootcamps, los módulos que componen un bootcamp, los alumnos apuntados a un bootcamp concreto. También queremos saber el profesor que va a impartir el modulo en ese bootcamp

Necesitamos poder después sacar este tipo de información de la base de datos:

- Alumnos apuntados a un bootcamp con nombre, apellidos, NIF, Teléfono (solo 1), email, Sexo, fecha nacimiento.
- Sacar los módulos que tiene un alumno en el bootcamp y la nota que tiene puesta, fecha de corrección proyecto
- Listado de profesores de un bootcamp, con sus datos personales, el nombre del bootcamp

Ejercicio 2 – Solución

Queremos gestionar los **alumnos** de Keepcoding. Para ello queremos tener los alumnos, los **bootcamps**, los **módulos** **que componen un bootcamp**, los alumnos apuntados a un bootcamp concreto. También queremos saber el **profesor** que va a impartir el modulo en ese bootcamp

Necesitamos poder después sacar este tipo de información de la base de datos:

- Alumnos apuntados a un bootcamp con nombre, apellidos, NIF, Teléfono (solo 1), email, Sexo, fecha nacimiento.
- Sacar los módulos que tiene un alumno en el bootcamp y la nota final puesta.
- Listado de profesores de un bootcamp, con sus datos personales, el nombre del bootcamp



Desnormalización

- Es lo contrario a normalizar.
- Es pasar un modelo normalizado correctamente, a un modelo sin normalizar.
- ¿Para que se usa eso?
 - Temas de Big data, cuadros de mando, ETLs etc.

FIN DE MODELADO

- Hasta aquí ya hemos modelado, analizado y verificado que el modelo esta normalizado y cumple los requisitos funcionales que nos han transmitido.
- ¿Y ahora? Pues aprenderemos SQL para poder crear la estructura del modelos (tablas , campos, tipos de datos, relaciones, FK, PK etc.) y después como añadir datos, buscar datos, eliminar datos, actualizar datos (Ciclo CRUD) usando siempre SQL.
- SQL se usa en la mayoría de las tecnologías (Swift, Kotlin, .NET, Java, NodeJS etc.)

Ejercicio apoyo 1

Tenemos en KeepCoding una biblioteca de **libros** de diferentes **temas** (tecnología, empresa y autoayuda). Queremos poder controlar los libros así como a que **alumnos** se les **entrega** los libros.

Ejercicio apoyo 2

Dato el modelo de cursos que vimos en clase.

Modificar el modelo para que se pueda tener un control de la nota que ha sacado cada **alumno** en cada **modulo**. Se quiere poder saber de un alumno y **bootcamp**, de cada modulo cuando entregó la practica y la nota de la practica (siendo apto o no apto los posibles valores de ese atributo).



KEEPCODING

Tech School

Madrid | Barcelona | Bogotá

Jose Luis Bustos Esteban
bejl@outlook.es