Módulo SQL – Base de datos

Contenidos:

- Introducción
- Diseño de bases de datos:
  - Requerimientos
  - Modelo Entidad-Relación
  - Modelo Relacional (Paso a tablas)
  - Implementación (Creación de tablas)
- Tratamiento de datos y consultas
  - Modificar Información
  - Consulta de registros
  - Consultas con varias tablas

Introducción

Introducción:

Sistema de información:

Un Sistema de Información es un conjunto de elementos destinados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo).

Esta definición de *Sistema de Información* abarca cualquier tipo de sistema, tengan o no componentes informáticos o se apoyen en ellos. En este caso, nuestro sistema de información elegido va a ser una base de datos, y vamos a trabajar utilizando un Sistema Gestor de Bases de Datos (MySQL) para analizarla y modificarla a nuestro gusto.

https://www.lavanguardia.com/internacional/20201005/483861404090/alarma-reino-unido-recuento-fallo-tecnico-contagios.html

Introducción:

Base de datos:

Se llama base de datos, o también banco de datos, a un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y/o transmisión.

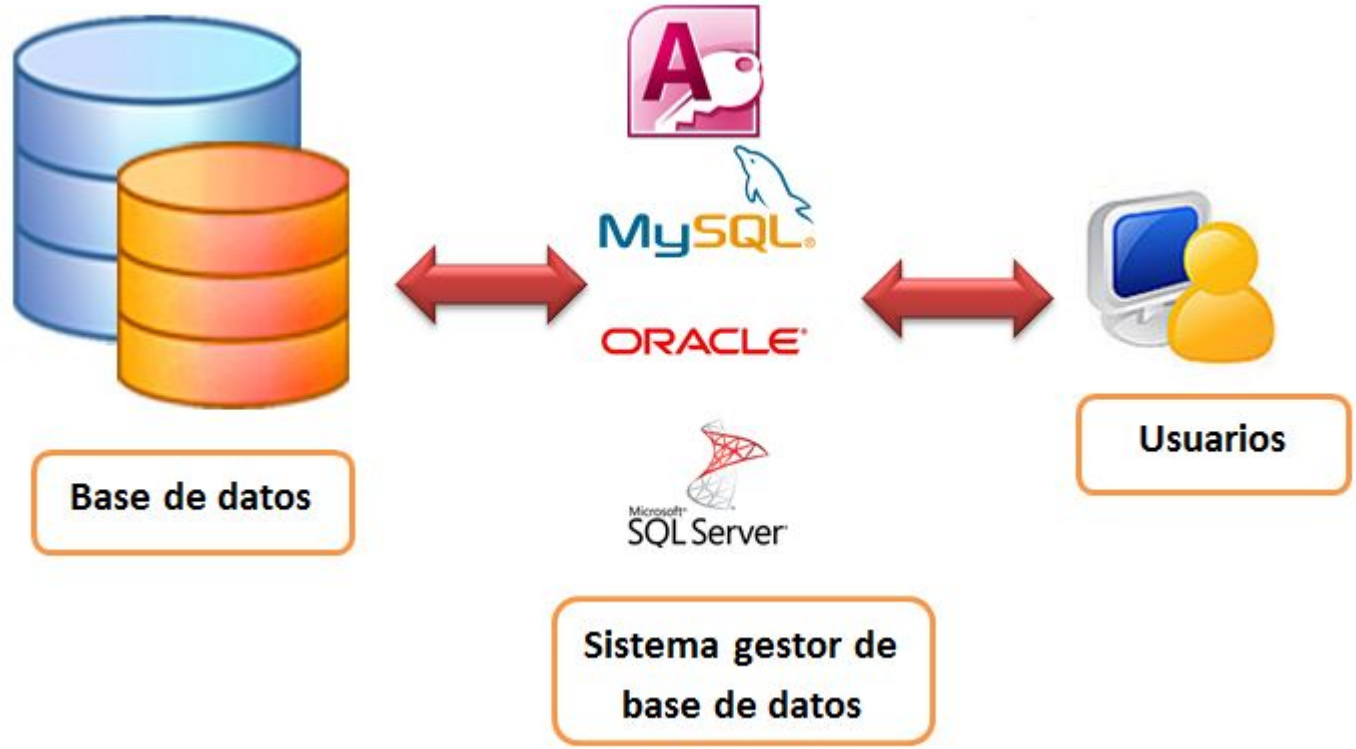
Introducción:

Sistemas Gestores de Bases de Datos (SGBD):

Un SGBD es una herramienta de software que permite la creación y gestión de una Base de Datos, organizando sus ficheros manteniendo la información siempre accesible para el usuario de la forma más eficiente posible, tanto en espacio como en velocidad de acceso.

Importante! Gestionando una Base de Datos a través de un SGBD, estamos en un nivel de abstracción superior, pues el acceso a los datos lo realizamos con un intermediario (SGBD).

Introducción:



Introducción:

Instalación MySQL en Ubuntu:

1. Actualizar paquetes disponibles:

En terminal:

sudo apt-get update (Descargará ficheros necesarios del sistema)

sudo apt-get upgrade (Actualizamos los paquetes instalados en el pc)

Introducción:

Instalación MySQL en Ubuntu:

2. Instalar MySQL Server:

En terminal:

sudo apt-get install mysql-server mysql-client (Instalación de MySQL Server)

Se utilizarán 245 MB de espacio de disco adicional después de esta operación. ¿Desea continuar? [S/n]

Pulsamos S para indicarle que si.

Introducción:

Instalación MySQL en MacOS:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"

brew install mysql

brew tap homebrew/services

brew services start mysql

Introducción:

Checkpoint:

mysql --version

Si el proceso de instalación ha ido bien se debería mostrar una línea similar a:

mysql Ver 8.0.22-0ubuntu0.20.10.2 for Linux on x86_64 ((Ubuntu))

Si no da error todo va bien.

Introducción:

Checkpoint:

mysql --version

Si el proceso de instalación ha ido bien se debería mostrar una línea similar a:

mysql Ver 8.0.22-0ubuntu0.20.10.2 for Linux on x86_64 ((Ubuntu))

Si no da error todo va bien.

Introducción:

3. Establecer configuración de seguridad inicial:

MySQL tiene un usuario de administración que se llama root.

Lo primero que vamos a hacer es asignarle como contraseña root y ajustar la configuración básica de seguridad

En un terminal ejecutar

sudo mysql_secure_installation

Lo primero que nos preguntará el asistente es si deseamos usar contraseñas seguras, en un entorno real debería activarse pero nosotros le vamos a decir que no porque vamos a usar una constraseña muy sencilla.

Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: n

Introducción:

3. Establecer configuración de seguridad inicial:

A Continuación nos solicita qué contraseña queremos asignarle al usuario root, le indicaremos el mismo nombre para la contraseña (root). Recordad que aunque no se vea lo que escribís (por seguridad) si que estáis introduciendo la contraseña.

Please set the password for root here.

New password:

Re-enter new password:

Introducción:

3. Establecer configuración de seguridad inicial:

Finalmente nos relizará unas preguntas sobre seguridad, le diremos a todo que sí: y

Remove anonymous users: y

Disallow root login remotely: y

Remove test database and access to it: y

Reload privilege tables now: y

Al finalizar el asistente se mostrará por pantalla All done! y nos devolverá al terminal.

Introducción:

4. Permisos de acceso

Desde un terminal ejecutar:

mysql -u root -p (acceso a mysql desde terminal)

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'root'; (modificar la password de root)

GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION; (permitir accesos)

FLUSH PRIVILEGES; (ejecución de orden de permisos)

Introducción:

5. Creación de usuario personal

Desde un terminal ejecutar:

mysql -u root -p (acceso a mysql desde terminal)

CREATE USER 'demo'@'localhost' IDENTIFIED BY 'password'; (crear usuario y contraseña)

GRANT ALL PRIVILEGES ON *.* TO demo@'localhost' WITH GRANT OPTION; (permitir accesos)

FLUSH PRIVILEGES; (ejecución de orden de permisos)

exit; (salimos de mysql)

mysql -u demo –p (comprobación de que funciona correctamente)

Introducción:

6. Instalación de MySQL Workbench:

Abrimos un terminal:

sudo snap install mysql-workbench-community

sudo snap connect mysql-workbench-community:password-manager-service :password-manager-service

Comprobar que se ha instalado el MySQL Workbench

Introducción:

7. Configuración de MySQL Workbench y conexión a la base de datos:

Iremos desde, el escritorio de Linux, a: Aplicaciones, Seleccionaremos Todas (en las parte de abajo), y pulsaremos sobre MySQL Workbench. Esto abrirá una aplicación de escritorio.

Pulsaremos en el + que está a la derecha de MySQL Connections y cubriremos con los siguientes datos

Conection name: Example

Hostname: 127.0.0.1

Port: 3306

User: demo

Password: (pulsamos en el botón Store in Keychain y la introducimos)

Finalmente pulsaremos sobre Test connection. y si todo ha ido bien se mostrará una ventana emergente que pondrá Successfully made the MySQL connection

Introducción:

8. Desinstalación:

sudo apt-get remove --purge mysql-server mysql-client mysql-common -y sudo apt-get autoremove -y

sudo apt-get autoclean

rm -rf /etc/mysql

rm -rf /var/lib/mysql

CUIDADO: No ejecutar si no se está seguro

Introducción:

8. Desinstalación:

En ocasiones pueden quedar ficheros relacionados con MySQL después de la desinstalación.

Con el siguiente comando localizaremos cualquier fichero que comience por mysql. OJO: ¡Puede haber ficheros que usan otras aplicaciones que comiencen por mysql! sudo find / -iname 'mysql*'

PELIGRO: Sólo ejecutar si realmente estamos seguros de lo que hacemos.

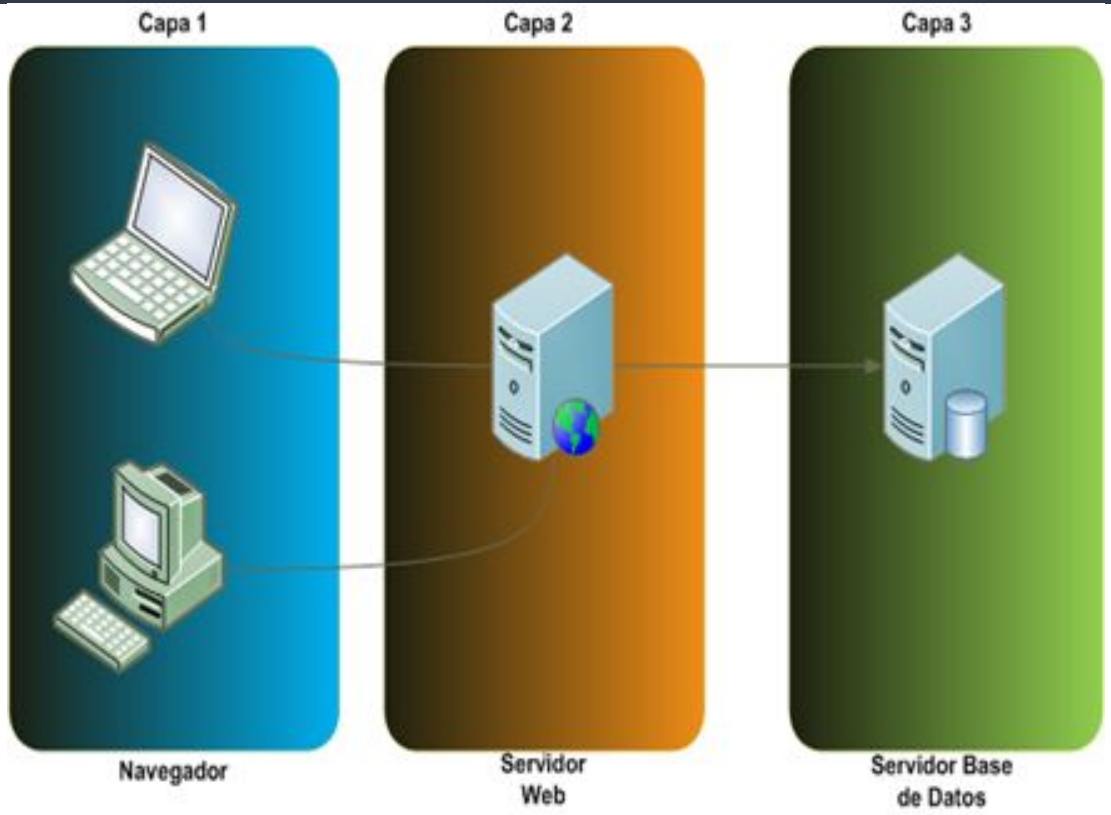
Con esta sentencia eliminamos todos los resultados anteriores. Nota: Si solamente queremos borrar unos ficheros en particular tendremos que eliminarlos manualmente.

sudo find / -iname 'mysql*' -exec rm -rf {} \;

Introducción:

Arquitectura de una aplicación:

Introducción:



Introducción:

Introducción:

Características deseables:

- CRUD: Create, Read, Update, Delete. Es decir, podremos crear registros, leerlos, actualizarlos y borrarlos.
- Correcta ejecución de transacciones - ACID: Atomicity, Consistency, Isolation, Durability.
  - Atomicidad: Se debe asegurar si la operación se ha ejecutado o no.
  - Consistencia: Una operación no pondrá en riesgo la integridad de la base de datos
  - Aislamiento: Si se ejecutan dos transacciones sobre los mismos datos al mismo tiempo, deben ser independientes y no generar error.
  - Durabilidad: Una vez se realice la operación, ésta persistirá en el tiempo.
- Recuperación: Obtener información en base a unos filtros.
- Validez: Validación de datos

Introducción:

Base de datos RELACIONAL:

- Tipos de datos: Cada columna tiene un tipo de dato definido de manera que el SGBD no permite almacenar valores de otro tipo en dicha columna
- Restricciones: Es posible definir restricciones que obliguen a cumplir una serie de requisitos a los valores que se almacenen en una columna determinada
- Integridad referencial: En el momento de registrar algún nuevo dato que deba estar relacionado con otro, el SGBD comprobará que el segundo existe antes de permitir el registro. En caso contrario no lo permitirá
- Consultas complejas: Es posible realizar consultas muy complejas, incluso aquellas que inicialmente no habían sido tenidas en cuenta durante el diseño de la Base de Datos

Diseño de base de datos

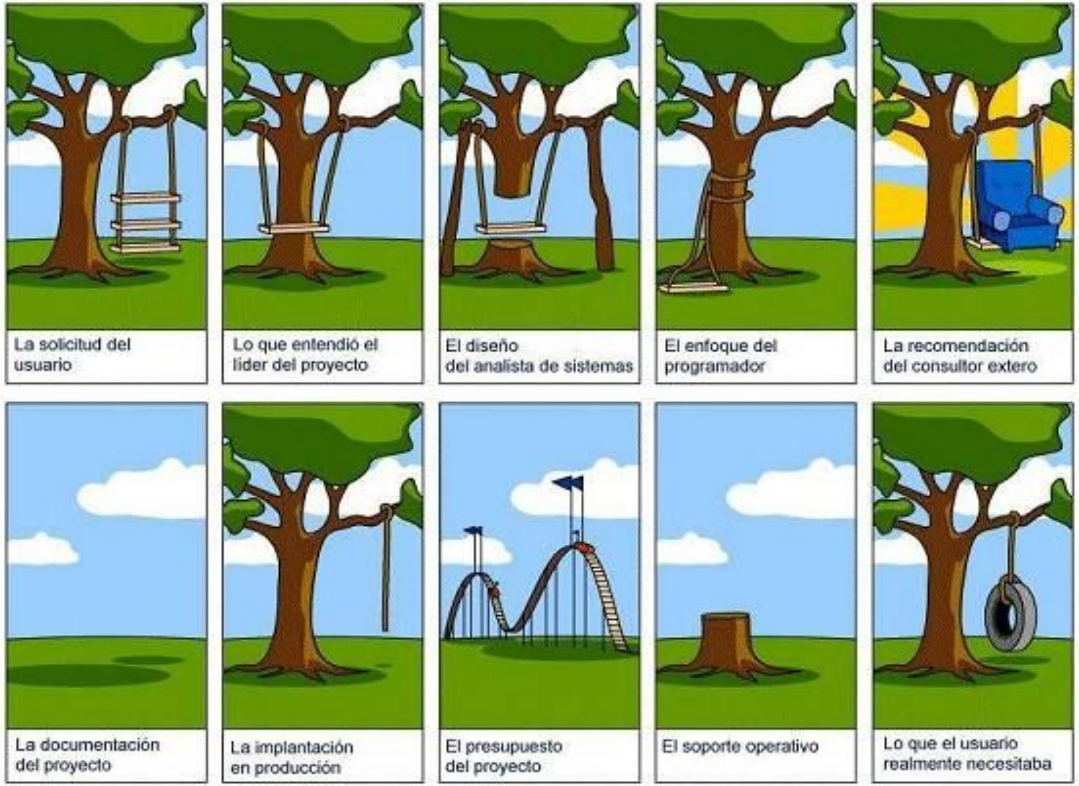
Diseño de base de datos:

1.Recolección y análisis de requerimientos:

En este paso recogemos información del sistema para el que debemos diseñar la Base de Datos:

- Determinar cuál es nuestro dominio de interés (minimundo)
- Determinar cuáles son sus límites
- Definir exactamente cuál va a ser el uso
- Obtener diferentes puntos de vista
- Anticiparnos a posibles problemas

Diseño de base de datos:



Diseño de base de datos:

2. Modelo Entidad-Relación

Es un modelo de datos que representa la realidad a través de entidades , que son objetos que existen y se distinguen de otros por sus características, que llamamos atributos. Además, estas entidades podrán o no estar relacionadas unas con otras a través de lo que se conoce como relación.

Hay que tener en cuenta que se trata solamente de un modelo de representación, por lo que no tiene correspondencia real con ningún sistema de almacenamiento. Se utiliza en la etapa de Análisis y Diseño de una Base de Datos, por lo que habrá que convertirla a tabla antes de poder empezar a trabajar con ella.

Diseño de base de datos:

Entidad

Una entidad es un objeto que existe en una realidad que queremos representar, por ejemplo, un alumno, que se distingue de otro por sus características como pueden ser: el nombre, los apellidos, el número de expediente,...

Las entidades se representan por el siguiente símbolo:

Diseño de base de datos:

Atributos

Esas características que hacen que unas entidades se distingan de otras, son los atributos. El nombre, los apellidos y el número de expediente serían atributos de la entidad alumno. Los atributos se representan por el siguiente símbolo:

Diseño de base de datos:

Relación

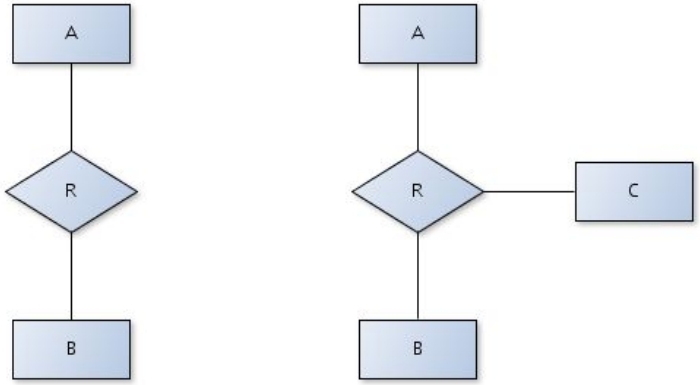
A su vez, podemos relacionar unas entidades con otras a través de lo que se conoce como relación. Por ejemplo, dos entidades alumno y asignatura podrían estar relacionadas entre sí puesto que un alumno cursa una asignatura (o varias). Conviene resaltar que una relación entre dos entidades no expresa obligatoriedad de relación sino posibilidad de relacionarse.

En este caso, no será necesario que todos los alumnos cursen una asignatura o que una asignatura sea cursada por todos los alumnos para que la relación se establezca. Por tanto, en este caso se establece que entre esas dos entidades existe una relación a la que podríamos llamar cursa. Las relaciones se representan por el siguiente símbolo:

Diseño de base de datos:

Cardinalidad de la relación:

Si consideramos que dos entidades A y B están relacionadas a través de una relación R, deberemos determinar lo que se conoce como cardinalidad de la relación, que determina cuantas entidades de tipo A se relacionan, como máximo, con cuantas entidades de tipo B. Además, resulta conveniente, en cada caso, calcular cuántas entidades de tipo A se relacionan, cómo mínimo, con cuantas entidades de tipo B (que normalmente será 0 ó 1). De esa manera podremos indicar la obligatoriedad o no de relación entre elementos de las entidades A y B.



Diseño de base de datos:

Relación uno a uno

En esta relación una entidad de tipo A sólo se puede relacionar con una entidad de tipo B, y viceversa. Por ejemplo, si suponemos dos entidades Curso y Aula, relacionadas a través de una relación Se Imparte, podremos suponer que un Curso se imparte en una Aula y en una Aula sólo se puede impartir un Curso. Se representaría como sigue:

Diseño de base de datos:

Relación uno a muchos

Indica que una entidad de tipo A se puede relacionar con un número indeterminado de entidades de tipo B, pero a su vez una entidad de tipo B sólo puede relacionarse con una entidad de tipo A. Si suponemos una entidad Propietario y otra entidad Vehículo relacionadas a través de una relación Posee, podremos suponer que un Propietario puede poseer varios Vehículos, mientras que cada Vehículo sólo puede pertenecer a un Propietario.

Quedaría representado de la siguiente manera:

Diseño de base de datos:

Relación muchos a muchos

En este caso, tanto las entidades de tipo A y B, pueden relacionarse con un número indeterminado de entidades del otro tipo. Por ejemplo, si suponemos las entidades Alumno y Asignatura y una relación Cursa, podremos suponer que un Alumno cursa varias asignaturas mientras que una Asignatura la cursan varios Alumnos.

Quedaría representado de la siguiente manera:

Diseño de base de datos:

Ejercicios:

- Un país y sus poblaciones.
- Almacenes que guardan distintos productos. Puede ser que algún almacén esté vacío. Los productos siempre van a estar en almacenes.
- Canciones y discos. Un disco siempre tendrá más de una canción, pero puede haber singles, es decir, canciones que no se agrupen en discos.
- En el mundo de los ordenadores hay muchas marcas. Cada uno de los modelos de ordenadores está asociado a una marca. Y todas las marcas tienen algún modelo.
- Una persona siempre va a estar identificado por un DNI, salvo que se trate de un bebé o niño. Los DNIs siempre tienen dueño.
- Hay diversos tipos de café, que se componen de al menos un tipo de grano. Los granos siempre forman parte de al menos un tipo de café.

Diseño de base de datos:

Ejercicios:

- Existen marcas que poseen distintos tipos de coche (sedán, monovolumen…). Sin embargo, pueden existir tipos de coche descatalogados.
- Los posts de un blog suelen estar etiquetados bajo múltiples etiquetas. Las etiquetas pueden o no estar asociadas a múltiples posts.
- En un barrio de una ciudad se puede dar el caso de que existan o no farmacias. Eso sí, las farmacias sólo estarán asociadas a un barrio en concreto.
- Un procesador puede o no ejecutar múltiples tareas. Las tareas son ejecutadas por un único procesador. También pueden estar a la espera de ser asignadas a uno de ellos.
- Existen trabajadores en una empresa. Si estos no cumplen el código de conducta, se les creará un único expediente donde se almacenarán todas las ofensas. Los expedientes son únicos e intransferibles.

Diseño de base de datos:

Ejercicios:

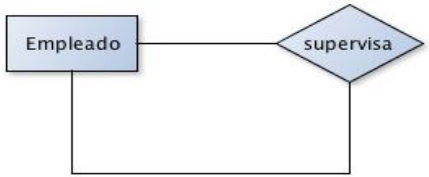
- Las respuestas sólo pueden estar enlazadas a una pregunta, pero una pregunta puede o bien no tener respuesta, o bien puede ser resuelta de distintas maneras.
- Las patologías de una enfermedad, pueden tener cura (o no) gracias a unos medicamentos. A su vez, los medicamentos pueden tratar diversas patologías.
- Los camareros pueden trabajar en distintos bares. Puede haber a su vez, camareros en paro. Los bares necesitan de al menos un camarero para estar en funcionamiento. Además, los bares pueden dividirse en mesas (estas no pueden estar en dos bares distintos). Eso sí, puede darse el caso, que algún bar no tenga mesas.
- Existen comentarios de post, que pueden o no hacer referencia a otros comentarios de post. Un comentario de post puede tener múltiples comentarios de posts que le referencian.

Diseño de base de datos:

Reflexividad:

Es posible que la misma entidad ocupe ambos lados de una relación. En ese caso estamos frente a lo que se conoce como relaciones reflexivas. La cardinalidad de la relación indicará si todos los elementos de la relación están relacionados reflexivamente o bien sólo algunos están relacionados entre sí.

En el caso de la figura podríamos suponer una empresa en la que algunos empleados hacen de supervisor de otros



Diseño de base de datos:

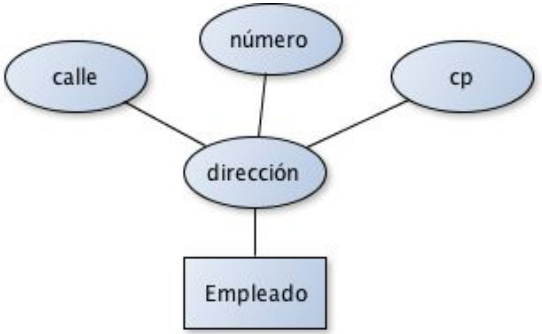
Atributos multivaluados:

Los atributos multivaluados son aquellos atributos que pueden contener una cantidad indeterminada de valores.

Diseño de base de datos:

Atributos estructurados (o compuestos)

Los atributos estructurados o compuestos son aquellos atributos que pueden estar compuestos por otros atributos. Normalmente son atributos que pueden descomponerse aunque dependiendo del contexto de la aplicación puede no interesar hacer esa descomposición y tratarlo como un atributo simple.



Diseño de base de datos:

Comprobaciones sobre el Diagrama Entidad-Relación

- Comprobar que nuestro diagrama no se ha convertido en un diagrama de flujo y no describe procesos, sino almacenes de datos
- Comprobar que las Entidades son nombres de cosas y las relaciones son verbos
- Comprobar que ninguna Entidad tiene como atributo algo que existe como Entidad (si ocurre, se deberían relacionar ambas Entidades)
- Comprobar que varias entidades no comparten un mismo atributo estructurado que pueda ser considerado realmente como una Entidad
- Evitar los ciclos (si aparece alguno, que puede ocurrir, comprobar que es necesario)
- Si una relación tiene varios atributos, valorar si es posible que realmente deba ser una nueva Entidad (Comprar → Pedido, Alquilar → Alquiler, Reservar → Reserva, Enviar → Envío, . . .)
- Comprobar que no hay colocada ninguna cardinalidad al revés: Se tiene que poder leer: un Usuario Realiza de 0 a N Pedidos. Usuario y Pedido son entidades y Realizar la relación entre ambas. En este caso, (0, N) debería estar escrito en el lado Pedido para que pudiera leerse correctamente

Diseño de base de datos:

Ejercicio 1:

- alumnos que cursan asignaturas, las cuales son impartidas por un profesor
- las asignaturas a su vez, pertenecen a un curso, el cual tiene designada un aula dónde se imparte la clase.
- los alumnos y profesores tienen los atributos de nombre y apellidos
- las asignaturas tienen los atributos de de nombre y nº de horas
- los cursos tienen los atributos de turno y código
- las aulas tienen los atributos de número y capacidad

Diseño de base de datos:

Ejercicio 2:

Se quiere diseñar una Base de Datos para almacenar todos los datos de un campeonato de fútbol sala que se organiza este año en la ciudad. Aquellos que quieran participar deberán formar un equipo (nombre, patrocinador, color de la 1ª camiseta, color de la 2ª camiseta, categoría, . . .) e inscribirse en el campeonato.

A medida que transcurran los partidos se irán almacenando los resultados de éstos, así como qué equipos lo jugaron, en qué campo se jugó, quién lo arbitró y alguna incidencia que pudiera haber ocurrido (en caso de que no ocurran incidencias no se anotará nada. Además, los participantes deberán rellenar una ficha de suscripción con algunos datos personales (nombre, apellidos, edad, dirección, teléfono, . . .)

Diseño de base de datos:

Ejercicio 3:

Se quiere diseñar una Base de Datos para controlar el acceso a las pistas deportivas de Zaragoza. Se tendrán en cuenta los siguientes supuestos:

- Todo aquel que quiera hacer uso de las instalaciones tendrá que registrarse y proporcionar su nombre, apellidos, email, teléfono, dni y fecha de nacimiento
- Hay varios polideportivos en la ciudad, identificados por nombre, dirección, extensión (en m2)
- En cada polideportivo hay varias pistas de diferentes deportes. De cada pista guardaremos un código que la identifica, el tipo de pista (tenis, fútbol, pádel, . . .), si está operativa o en mantenimiento, el precio y la última vez que se reservó
- Cada vez que un usuario registrado quiera utilizar una pista tendrá que realizar una reserva previa a través de la web que el ayuntamiento ha creado. De cada reserva queremos registrar la fecha en la que se reserva la pista, la fecha en la que se usará y el precio. Hay que tener en cuenta que todos los jugadores que vayan a hacer uso de la pista deberán estar registrados en el sistema y serán vinculados con la reserva

Diseño de base de datos:

Ejercicio 4:

Una empresa desea crear un sitio Web de comercio electrónico al que se podrán conectar clientes para realizar sus compras. Se tiene que realizar el diseño de la Base de Datos que soporte la operativo de este sitio Web.

- Cuando un usuario intenta entrar en este sitio, se le pedirá un login y una contraseña. El sistema comprobará si el usuario tiene cuenta y en caso negativo se le pedirán los siguientes datos de alta: NIF, correo, nombre, dirección, teléfono, login y password. Se comprobará si ya existía con distinto login para darle un mensaje de error.
- Una vez el usuario se ha dado de alta o ha entrado con su login y password correctos, puede visitar las distintas secciones de la tienda virtual. Nuestra empresa quiere que quede constancia de las secciones visitadas por los usuario y la fecha en la que la visitaron. Hay que tener en cuenta que un usuario podrá visitar varias secciones. De cada sección se almacenará un código, nombre, descripción y fecha de creación. (prosigue...)

Diseño de base de datos:

Ejercicio 4:

- Los usuarios pueden realizar sus compras utilizando un carrito virtual. Cuando un usuario decide utilizar el carrito, el sistema creará uno almacenando la fecha de creación. El usuario entonces puede poner productos, detallando cuantas unidades desea o bien eliminarlos. Un carrito puede contener varios productos y un producto puede aparecer en carritos de diferentes usuarios.
- De los productos se almacenará el código de producto, el nombre, la descripción y el precio por unidad. Cuando un usuario decide finalizar su compra, el sistema le pedirá entonces los datos bancarios (si es la primera vez que paga) y dará el carrito por finalizado. El usuario puede dejar un carrito lleno y no completar la compra en esa sesión, para completarla otro día. El usuario debe poder comprobar cuál es el coste total de un carrito antes de pagarlo. Además podrá comprobar el precio total de todos sus carritos anteriores y su contenido.
- En este sitio Web los productos están organizados en las diferentes secciones teniendo en cuenta que un producto puede aparecer en varias secciones y una sección puede tener varios productos

Diseño de base de datos:

3. Modelo Relacional (Paso a tablas):

- Toda entidad se transforma en una tabla
- Todo atributo simple o derivado se transforma en columna de una tabla
- Los atributos estructurados transforman los campos en los que se componen en nuevas columnas de la tabla
- El identificador único de la entidad se convierte en clave primaria
- Los atributos multivaluados generan una nueva tabla con tres columnas: un id, el id de la tabla de la que surgen propagado como clave ajena y el valor del campo multivaluado

Diseño de base de datos:

Paso a tablas: Relaciones

- Toda relación N:M se transforma en una tabla que tendrá como clave primaria la concatenación de los atributos clave de las entidades que relaciona.
- En la transformación de relaciones 1:N existen dos posibilidades:
  - Transformarlo en una tabla. Se hace como si se tratara de una relación N:M. Es conveniente realizarlo así en el caso de que se prevea que en un futuro la relación puede transformarse en N:M y cuando la relación tiene atributos propios
  - Propagar la clave. Se propaga el atributo principal de la entidad que tiene de cardinalidad máxima 1 a la que tiene cardinalidad máxima N, haciendo desaparecer a la relación

Diseño de base de datos:

Paso a tablas: Relaciones

- En la transformación de relaciones 1:1:
  - Propagar la clave. Si una de las entidades posee cardinalidad (0,1) y la otra (1,1), se propaga la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad de cardinalidad (0,1). Si ambas poseen cardinalidades (1,1), se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra
- Para los atributos de las relaciones existen dos casos:
  - Si la relación es 1:N, sus atributos se propagan a la tabla de lado N, junto con la clave del lado 1
  - Si la relación es N:M, sus atributos se transforman en columnas de la tabla generada por dicha relación