

1. ¿Qué es una base de datos?

- Una base de datos es un conjunto de ficheros o tablas relacionadas entre sí, cuyos datos pueden ser compartido entre los usuarios con una integridad máxima y una redundancia mínima.

2. ¿Qué significa que las bases de datos tienen que tener una *integridad máxima*?

- Integridad máxima: los datos procesados han de ser única y exclusivamente los que desea el usuario. De la misma forma los cambios realizados en una tabla han de tener en cuenta las reglas anteriores.

3. ¿Y *redundancia mínima*?

- Redundancia mínima: evitar que los datos se repitan innecesariamente.

4. Busca una página web que utilice al menos una base de datos para mostrar la propia página web.

- Reddit → www.reddit.com/

5. ¿Qué hardware, software y usuarios son necesarios para diseñar/crear una base de datos?

- El hardware necesario para crear una base de datos consiste en un sistema informático que contenga la capacidad de almacenamiento físico suficiente para almacenar los datos.

- El software necesario para crear una base de datos consiste en un sistema operativo y un programa (SGB) que permita crear y diseñar la estructura lógica de la base de datos.

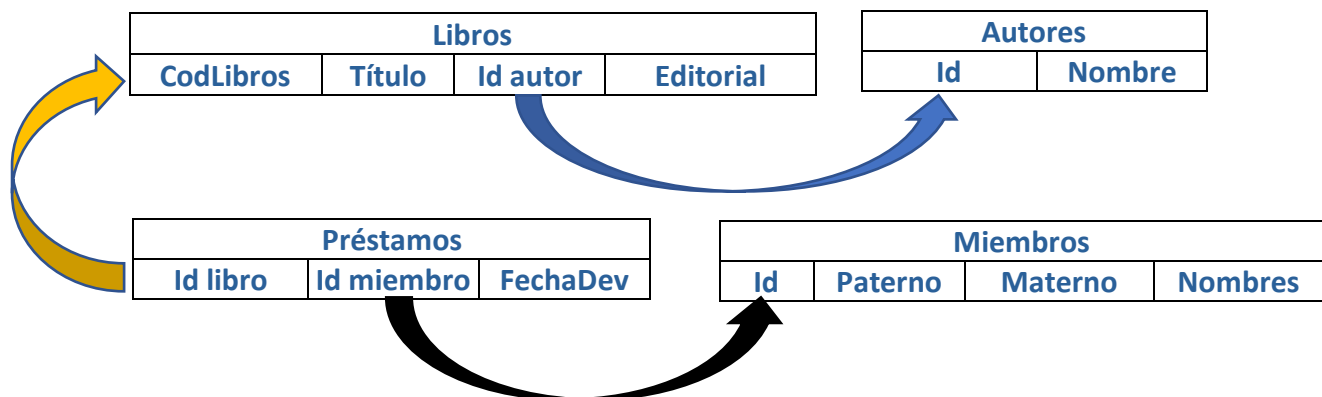
6. Haz dos tablas que representen una correcta Integridad Referencial. Explica un posible error de Integridad Referencial.

Miembros gimnasio				Check-in gimnasio			
Id	Nombre	Apellidos	Edad	Id miembro	Día	Hora entrada	Hora salida

- En dichas tablas podría ocurrir un error de integridad referencial si algún miembro del gimnasio se da de baja, ya que los registros de la tabla "check-in gimnasio" podrían no hacer referencia a ningún miembro de la tabla "miembros gimnasio" si no hay unas reglas bien definidas.

7. ¿Cómo solucionarías este problema de redundancia de datos?

<i>CodLibro</i>	<i>Título</i>	<i>Autor</i>	<i>Editorial</i>	<i>Paterno</i>	<i>Materno</i>	<i>Nombres</i>	<i>FechaDev</i>
1001	Variable compleja	Murray Spiegel	McGraw Hill	Pérez	Gómez	Juan	15/04/2005
1004	Visual Basic 5	E. Petroustos	Anaya	Ríos	Terán	Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGraw Hill	Roca		René	16/04/2005
1006	Oracle University	Nancy Greenberg	Oracle Corp.	García	Roque	Luis	20/04/2005
1006	Oracle University	Priya Nathan	Oracle Corp.	García	Roque	Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGraw Hill	Pérez	Gómez	Juan	18/04/2005



- Para estructurarla de mejor manera, separaría la tabla en cuatro tablas (Libros, Autores, Miembros y Préstamos). Relacionaría la tabla préstamos con ambas tablas. Con la tabla "Libros" a través del campo "CodLibros" y con la tabla miembros a través del campo "id". La tabla "Libros" a su vez se relacionaría con la tabla "Autores"

8. ¿Por qué es necesario tener un control de la concurrencia en una base de datos?

- Es necesario tener un control de la concurrencia en una base de datos ya que si varios usuarios están accediendo al mismo dato al unísono este puede perder su integridad si varios los modifican u operan con él, por lo que el sistema debe bloquear que esto suceda, pero también es importante que ningún usuario quede desatendido.

9. ¿Qué tipo de relaciones existen según la cantidad de tablas involucradas? Pon ejemplos de cada tipo.

- Según la cantidad de tablas involucradas se pueden distinguir las siguientes relaciones:

- **Relación unaria o reflexiva:** se relaciona una tabla consigo misma. Por ejemplo: una tabla de personas donde se crea un campo de “mejor amigo” que hace referencia a “ids” (o DNIs) de otras personas de la misma tabla.
- **Relación binaria:** se relacionan dos tablas. Por ejemplo: una tabla de mapas se puede relacionar con el juego al que corresponde.
- **Relación ternaria:** se relacionan tres tablas. Por ejemplo: una tabla de dispositivos se puede relacionar al mismo tiempo con una tabla de ubicaciones (oficina A, piso 3, edificio Alpha...) y otra tabla de tipo de dispositivo (CPU, Intel, i5...)

10. ¿Qué tipo de relaciones existen según la cantidad de registros que intervienen en la relación? Pon ejemplos de cada tipo.

- Según la cantidad de registros que intervienen, una relación puede ser:

- **Relación uno a uno (1:1):** un registro de una tabla “A” solo se puede relacionar con un registro de una tabla “B” y viceversa. Por ejemplo: Un registro de la tabla personas tan solo se podría relacionar con un registro de la tabla becas y viceversa.
- **Relación uno a muchos (1:M):** un registro de una tabla “A” se puede relacionar con varios registros de una tabla “B”, sin embargo, un registro de la tabla “B” solo se puede relacionar con uno de la tabla “A”. Por ejemplo: Un registro de poblaciones se puede relacionar con varios registros de la tabla calles, sin embargo, un registro de la tabla calles tan solo se podría relacionar con un registro de la tabla poblaciones
- **Relación muchos a muchos (M:M):** un registro de una tabla “A” se puede relacionar con varios registros de una tabla “B” y viceversa. Por ejemplo: Un registro de la tabla comidas se puede relacionar con varios registros de la tabla ingredientes y viceversa.

11. ¿Qué es un SGBD?

- Un Sistema Gestor de Bases de Datos es un programa o una serie de programas que permiten gestionar la base de datos, permitiendo separar la estructura física y lógica de la base de datos, así como proporcionar a los usuarios los datos que necesitan.

12. ¿Cuáles suelen ser los lenguajes de trabajo de los SGBD? Investiga los comandos que utiliza SQL para cada lenguaje.

- Los lenguajes de trabajo en los SGBD se pueden clasificar en:

- **Lenguajes de definición (LDD):** utilizados para crear la base de datos o modificar su estructura
- **Lenguajes de control (LCD):** utilizado para dar o revocar permisos a los usuarios
- **Lenguajes de manipulación (LMD):** utilizados para procesar los datos, es decir, hacer consultas, insertar o eliminar datos, modificar los datos, etc.

- En SQL se utilizan los siguientes comandos para cada lenguaje

- **Lenguajes de definición (LDD):** CREATE, ALTER, DROP, ADD, TRUNCATE
- **Lenguajes de control (LCD):** GRANT, REVOKE
- **Lenguajes de manipulación (LMD):** SELECT, DELETE, UPDATE, FROM, JOIN, LIMIT, HAVING, COUNT, GROUP BY, BETWEEN, IN