

## SOFTWARE PARA SISTEMAS INFORMÁTICOS (VI)

1. Conceptos y definiciones básicas
  - 1.1 Bases de datos, Sistemas Gestores de Bases de datos, entidades y relación
  - 1.2 Modelos de bases de datos
2. Utilización de los gestores de bases de datos
  - 2.1.- Características principales de los sistemas de gestión de bases de datos
  - 2.2.- Ventajas y desventajas de los sistemas de gestión de bases de datos
3. Diseño de una base de datos
  - 3.1.- Diseño conceptual
  - 3.2.- Diseño lógico
  - 3.3.- Diseño físico
4. Los registros y los campos
5. Ordenación y selección de registros
6. Los filtros

### 1. Conceptos y definiciones básicas

#### 1.1 Bases de datos, Sistemas Gestores de Bases de datos, entidades y relación

**Base de datos:** Una base de datos es un archivo o conjunto de archivos que contienen múltiples informaciones que de alguna forma guardan relación. Por ejemplo, una base de datos para gestionar un videoclub almacenará múltiples informaciones sobre películas, socios, etc... y entre esas informaciones existirán relaciones como, por ejemplo, una película con un socio por medio de un préstamo o alquiler.

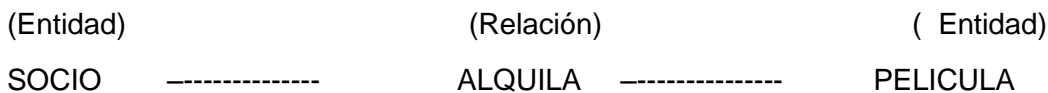
**Sistema Gestor de Bases de Datos (SGBD):** Para construir una base de datos en soporte informático con todas las informaciones a almacenar (estructuras de datos, tablas, índices, etc...), es necesario disponer de una herramienta que lo permita. Este es el cometido de un SGB, que básicamente permite crear, manipular, gestionar y eliminar tanto los datos como las estructuras de una base de datos, permitiendo de esta forma el tratamiento automatizado y fácil de la información almacenada en ella.

No se debe confundir SGBD con Base de Datos, el primero es una herramienta para la creación mientras que la segunda es una solución concreta a un problema de almacenamiento de datos determinado. Por ejemplo, con LibreOffice Base o con Microsoft Excel (dos ejemplos de SGBD) podemos crear y gestionar múltiples bases de datos (para gestionar un videoclub, una biblioteca, un comercio, etc...) con diferentes necesidades de almacenamiento.

**Entidad:** En líneas generales podemos entender por entidad como todo aquello sobre lo que es necesario almacenar información en una base de datos. Por ej. En una base de datos de una biblioteca entidades serían LIBROS, SOCIOS, etc....

**Relación:** Entre las diferentes entidades de una base de datos suele haber relaciones, las cuales permiten un tratamiento más globalizado de la información y reflejan situaciones del mundo real. Por ejemplo, en un videoclub hay dos entidades claras PELICULAS y SOCIOS. Entre ellas existe una relación bastante evidente que es el préstamo o alquiler.

De esta forma las entidades y relaciones en una base de datos se representan de forma lógica con estructuras del tipo:



Tanto el socio como el alquiler y la película, tienen sus informaciones asociadas que serán las que se almacenen en la base de datos.

Para la entidad socio, las informaciones podrían ser “número de socio”, “nombre y apellidos”, “número de teléfono”, etc.

Para la entidad alquiler las informaciones podrían ser “fecha”, “número de cinta”, “importe”, “número de socio”, etc.

Para la entidad película las informaciones podrían ser “número de cinta”, “título”, “director”, “género”, etc.

## 1.2 Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de Administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a Algoritmos, y Conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

### a) Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

### b) Bases de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

### c) Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad,

datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos.

#### **d) Base de datos relacionales**

Éste es el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas de las bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

#### **e) Bases de datos multidimensionales**

Son bases de datos ideadas para desarrollar aplicaciones muy concretas. Un ejemplo de este tipo de bases de datos es *Cubos OLAP*. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), pero muestran diferencias a nivel conceptual.

## **2. Utilización de los gestores de base de datos**

Como hemos visto antes, un gestor de base de datos o sistema de gestión de base de datos (SGBD o DBMS) es un software que permite introducir, organizar y recuperar la información de las bases de datos; en definitiva, administrarlas.

El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

### **2.1.- Características principales de los sistemas de gestión de bases de datos**

a) Abstracción de la información. Ahorran a los usuarios detalles acerca del almacenamiento físico de los datos.

b) Independencia. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

c) Redundancia mínima. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante.

d) Consistencia. Vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

e) Seguridad. Deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados.

f) Integridad. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados.

g) Respaldo y recuperación. Deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos.

h) Control de la concurrencia. Lo más habitual es que sean muchas las personas que acceden a

una base de datos; ésta debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

## 2.2.- Ventajas y desventajas de los sistemas de gestión de bases de datos

### a) Ventajas

La gran ventaja de utilizar un sistema de gestión de bases de datos es que el sistema provee facilidades para la manipulación de grandes volúmenes de datos como, por ejemplo, simplifican la introducción de datos, permiten manipularlos garantizando que los cambios sean consistentes y proveen de interfaces y lenguajes de consulta que simplifican la recuperación de los datos

2.- Simplifican la programación de equipos de consistencia.

3.- Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base serán siempre consistentes sin importar si hay errores correctamente, etc.

4.- Organizan los datos con un impacto mínimo en el código de los programas.

5.- Disminuyen drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.

6.- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

### DESVENTAJAS

1.- Típicamente, es necesario disponer de una o más personas que administren la base de datos, de la misma forma en que suele ser necesario en instalaciones de cierto porte disponer de una o más personas que administren los sistemas operativos. Esto puede llegar a incrementar los costos de operación en una empresa. Sin embargo hay que balancear este aspecto con la calidad y confiabilidad del sistema que se obtiene.

2.- Si se tienen muy pocos datos que son usados por un único usuario por vez y no hay que realizar consultas complejas sobre los datos, entonces es posible que sea mejor usar una hoja de cálculo.

3.- Complejidad: el software muy complejo y las personas que vayan a usarlo deben tener conocimiento de las funcionalidades del mismo para poder aprovecharlo al máximo.

4.- Tamaño: la complejidad y la gran cantidad de funciones que tienen hacen que sea un software de gran tamaño, que requiere de gran cantidad de memoria para poder correr.

5.- Coste del hardware adicional: los requisitos de hardware para correr un SGBD por lo general son relativamente altos, por lo que estos equipos pueden llegar a costar gran cantidad de dinero.

## 2.3.- Principales gestores de bases de datos

Nos centraremos en los gestores de bases de datos de tipo relacional, que son los genéricos disponibles en los paquetes de ofimática para usuarios de tipo doméstico.

Entre los gestores actuales más populares existen:

**MySQL.**

**PostgreSQL.**

**Oracle.**

**Microsoft SQL Server**

## HSQL

La aplicación LibreOffice Base y Apache OpenOffice Base utilizan el gestor HSQL (Hyperthreaded Structured Query Language). La aplicación Microsoft Access utiliza el gestor Microsoft SQL Server.

### 3.- Diseño de una base de datos en el modelo relacional

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de información determinado.

Para ello se siguen por regla general tres fases en el proceso de diseño: el diseño conceptual, el lógico y el físico.

En el diseño conceptual se hace una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD (Sistema Gestor de Bases de Datos) que se vaya a utilizar para manipularla. Su objetivo es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar dicha información.

El diseño lógico parte del resultado del diseño conceptual y da como resultado una descripción de la estructura de la base de datos. El diseño lógico depende del tipo de SGBD que se vaya a utilizar, se adapta a la tecnología que se debe emplear, pero no depende del producto concreto. En el caso de bases de datos convencionales relacionales (basadas en SQL para entendernos), el diseño lógico consiste en definir las tablas que existirán, las relaciones entre ellas, normalizarlas, etc...

El diseño físico parte del lógico y da como resultado una descripción de la implementación de una base de datos, es decir, las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Aquí el objetivo es conseguir una mayor eficiencia, y se tienen en cuenta aspectos concretos del SGBD sobre el que se vaya a implementar. Por regla general este proceso lo automatiza el SGBD y no lo manipula el usuario.

En el modelo relacional el diseño conceptual y el lógico se parecen mucho.

El diseño conceptual consiste en diagramas de Entidad/Relación. El diseño lógico consiste en tablas y relaciones entre esas tablas.

Este es el modelo utilizado por los sistemas gestores de datos más habituales (SQL Server, Oracle, MySQL...).

El modelo relacional de bases de datos se rige por algunas normas sencillas:

- Todos los datos se representan en forma de tablas, también llamadas "relaciones". La tabla es además la unidad de almacenamiento principal.
- Las tablas están compuestas por filas (o registros) y columnas (o campos) que almacenan la información sobre una entidad.
- Cada tabla debe poseer una clave primaria, esto es, un identificador único de cada registro compuesto por una o más columnas.
- Para establecer una relación entre dos tablas es necesario incluir, en forma de columna, en una de ellas la clave primaria de la otra. A esta columna se le llama clave externa. Ambos conceptos de clave son extremadamente importantes en el diseño de bases de datos.

Basándose en estos principios se diseñan las diferentes bases de datos relacionales, definiendo un diseño conceptual y un diseño lógico, que luego se implementa en el diseño físico usando para ello el gestor de bases de datos de nuestra elección (por ejemplo SQL Server).



producto, que será por regla general un número entero que se va incrementando cada vez que introducimos un nuevo producto (1, 2, 3, etc..).

Los campos marcados como "FK" son claves foráneas o claves externas. Indican campos que van a almacenar claves primarias de otras tablas de modo que se puedan relacionar con la tabla actual. Por ejemplo, en la tabla de productos el campo CategoryID está marcado como "FK" porque en él se guardará el identificador único de la categoría asociada al producto actual. En otras palabras: ese campo almacenará el valor de la clave primaria (PK) de la tabla de categorías que identifica a la categoría en la que está ese producto.

Los campos marcados con indicadores que empiezan por "I" (ej: "I1") se refieren a índices. Los índices generan información adicional para facilitar la localización más rápida de registros basándose en esos campos. Por ejemplo, en la tabla de empleados (Employees) existe un índice "I1" del que forman parte los campos Nombre y Apellidos (en negrita además porque serán también valores únicos) y que indica que se va a facilitar la localización de los clientes mediante esos datos. También tiene otro índice "I2" en el campo del código postal para localizar más rápidamente a todos los clientes de una determinada zona.

Los campos marcados con indicadores que empiezan con "U" (por ejemplo U1) se refieren a campos que deben ser únicos. Por ejemplo, en la tabla de categorías el nombre de ésta (CategoryName) debe ser único, es decir, no puede haber -lógicamente- dos categorías con el mismo nombre.

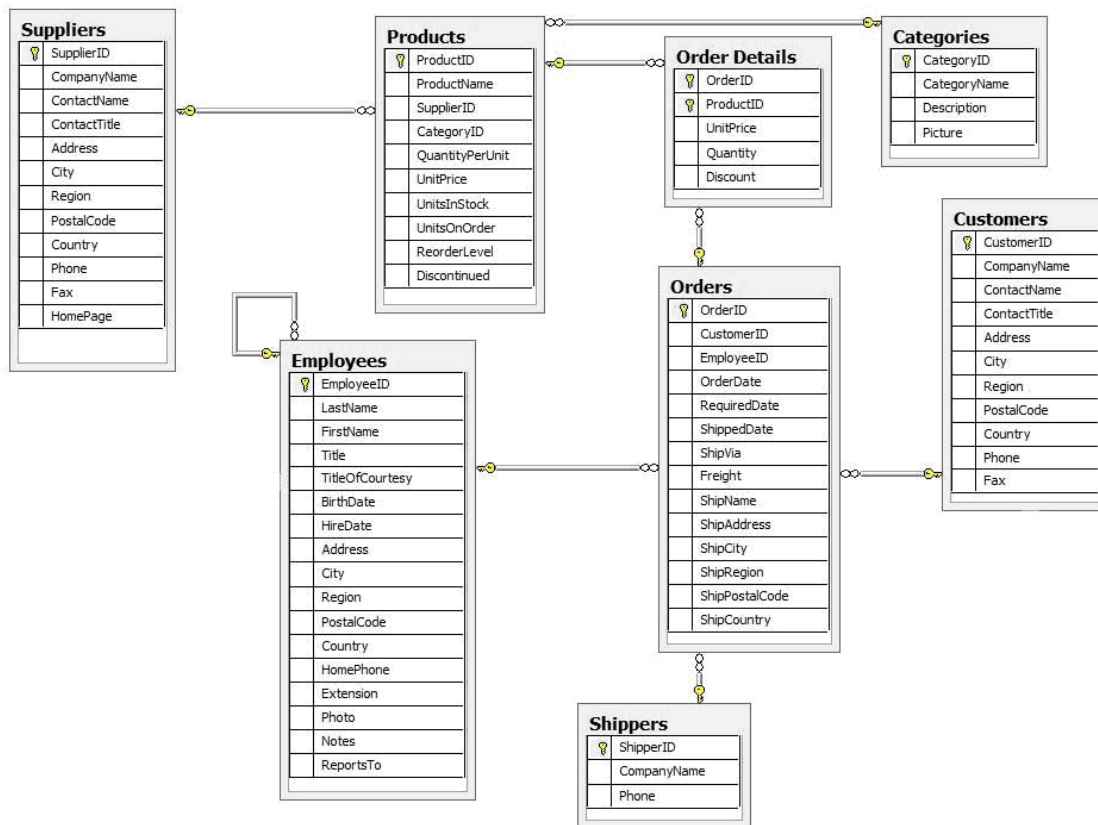
Como vemos, un diseño conceptual no es más que una representación formal y acotada de entidades que existen en el mundo real, así como de sus restricciones, y que están relacionadas con el dominio del problema que queremos resolver.

### 3.2) Diseño lógico

Una vez tenemos claro el modelo E-R debemos traducirlo a un modelo lógico directamente en el propio sistema gestor de bases de datos (Oracle, MySQL, SQL Server...). Si hemos utilizado alguna herramienta profesional para crear el diagrama E-R, seguramente podremos generar automáticamente las instrucciones necesarias para crear la base de datos.

La mayoría de los generadores de diagramas E-R (por ejemplo Microsoft Visio) ofrecen la capacidad de exportar el modelo directamente a los SGBD más populares.

Entonces, todo este modelo conceptual se traduce en un modelo lógico que trasladaremos a la base de datos concreta que estemos utilizando y que generalmente será muy parecido. Por ejemplo, este es el mismo modelo anterior, mostrado ya como tablas en un diagrama de SQL Server:



En este caso hemos creado cada tabla, una a una, siguiendo lo identificado en el diagrama E-R y estableciendo índices y demás elementos según las indicaciones de cada uno de los campos. Además hemos decidido el mejor tipo de datos que podemos aplicar a cada campo (texto, números, fechas... que se almacenan para cada registro).

Su representación gráfica en la base de datos es muy similar, sin embargo el modelo físico (cómo se almacena esto físicamente), puede variar mucho de un SGBD a otro y según la configuración que le demos.

Un buen diseño de base de datos debe poseer siempre las siguientes cualidades:

- Reflejar la estructura del problema en el mundo real.
- Ser capaz de representar todos los datos esperados, incluso con el paso del tiempo.
- Evitar el almacenamiento de información redundante.
- Proporcionar un acceso eficaz a los datos.
- Mantener la integridad de los datos a lo largo del tiempo.
- Ser claro, coherente y de fácil comprensión.

El primero, el diseño conceptual, es el que más tiempo nos va a llevar pues debemos pensar muy bien cómo vamos a representar las entidades del mundo real que queremos representar, qué datos almacenaremos, cómo los relacionaremos entre sí, etc...

El diseño lógico es mucho más sencillo puesto que no es más que pasar el diseño anterior a una base de datos concreta. De hecho muchas herramientas profesionales nos ofrecen la generación automática del modelo, por lo que suele ser muy rápido.



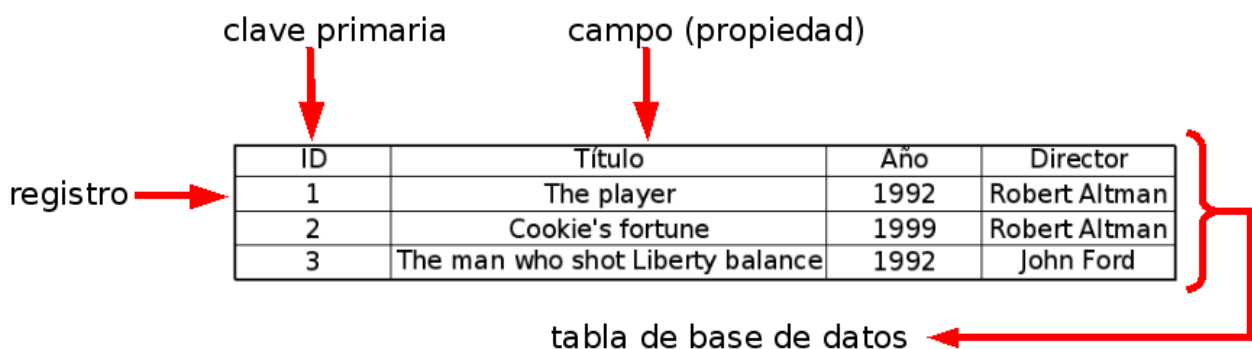
### 3.3) Diseño físico

El diseño físico por regla general recae en la propia base de datos, a partir del diseño lógico, aunque si dominamos bien esa parte elegiremos cuidadosamente índices, restricciones o particiones así como configuraciones para determinar cómo se almacenará físicamente esa información, en qué orden, cómo se repartirá físicamente en el almacenamiento, etc...

### 4.- Los registros y los campos

Aunque ya se ha mencionado en el apartado anterior, vale la pena redundar en los conceptos de tabla, campos y registros.

En las bases de datos relacionales la información se almacena en **tablas**, que ordenan los datos en filas y columnas. Las filas se denominan **registros** y las columnas se llaman campos. Los registros reciben también el nombre de **tuplas**.



Ejemplo de una base de datos para una agencia de alquiler de coches .Necesitaremos una tabla en la que se guarde información sobre los coches, como puede verse en la figura.

Marca	Modelo	Color	Matrícula	Situación
Lamborghini	Diablo 630	Amarillo	MA-2663-BC	En renta
Ferrari	F-40	Rojo	MA-8870-BC	Disponible
Sbarro R.	Decade	Blanco	VD-870-GTH	Disponible
De Tomaso	Pantera	Blanco	ML-7890-B	En renta
Pontiac	Trans-Am	Negro	KNIGHT	En taller
Austin M.	S3'40	Marrón	CA-5647-AB	Disponible
Jaguar	Destructor	Verde	AD-768-TTY	En renta

De esta forma, vemos que cada tabla está compuesta por filas, también llamadas tuplas o registros, cada uno de los cuales posee una serie de campos en los que se almacenan los datos básicos. El esquema de una tabla nos indica los nombres de cada uno de los campos que contiene, así como el tipo de información que debe contener. Una tabla es un conjunto de registros; por tanto, los registros no pueden repetirse. Para poder acceder a un registro concreto, es necesario hacer una consulta a través de algún campo que identifique a dicho registro, como

puede ser p.ej. el número de la matrícula. A este campo especial que identifica cada registro se le llama clave del registro.

La figura siguiente ilustra una tabla de clientes.

Apellidos	Nombre	D.N.I.	Edad
González Aranda	Javier	75836934	27
Beato Apóstol	Antonio	28836746	43
Campos Ortega	Adriano	82665358	36
Ruiz Rojo	Juan	83667228	35

¿Cómo nos las apañamos para indicar ahora qué cliente se hace responsable de cada coche alquilado? . Fácilmente, a través de una nueva tabla que relaciona los clientes con los coches.

Para ello dado que cada registro queda identificado por su clave, nos basta con incluir en esta nueva tabla a las claves de ambas tablas, en lugar de todos sus campos. Así, podemos obtener una nueva tabla de alquileres que contenga la matrícula del coche, y el D.N.I. del cliente, tal como se ve en la figura siguiente.

Matrícula	D.N.I.
MA-2663-BC	75836934
ML-7890-B	83667228
AD-768-TTY	75836934
AD-768-TTY	82665358

Este método de expresar los datos facilita a demás las consultas, que se realizan ahora a través de estas tablas especiales que relacionan a otras tablas. P.ej., si queremos saber los coches que ha alquilado González Aranda, basta con buscar su clave en la tabla de clientes (75836934), y a continuación ver que matrículas tiene asociadas en la tabla de alquileres (MA-2663-BC, y AD-768-TTY); a continuación, buscamos en la tabla de coches cuales son los coches que poseen esas claves, y obtenemos como resultado: Lamborghini y Jaguar.

Ejemplos de bases de datos muy simples son un directorio de personas o una lista de teléfonos: tenemos una única tabla organizada en tres o cuatro columnas/campos: nombre, dirección, teléfono, etc.

En las tablas grandes los registros pueden contener muchos campos. Depende de la cantidad de detalles que queramos almacenar acerca de un tema. El conjunto de datos de cada persona distinta en esa tabla, o sea, cada fila, se denomina registro. Las tablas pueden contener desde los pocos registros de nuestra agenda de teléfonos personal, hasta las decenas de millones de registros que gestionan las bases de datos de los bancos, compañías telefónicas, Hacienda, etc.

Las hojas de cálculo como LibreOffice Calc funcionan estupendamente como modestas bases de datos de una sola tabla. ¡Incluso una tabla escrita en LibreOffice Writer u otro procesador de textos puede considerarse como una primitiva base de datos!. En última instancia, siempre es preferible tener los datos sistemáticamente organizados de algún modo que carecer de organización

alguna. A la información almacenada en una base de datos se le llama información estructurada, ya que los datos se amoldan a una estructura preexistente.

Una sola tabla no basta cuando queremos mantener junta información sobre diferentes temas (entidades). Por ejemplo, llevar cuenta de los documentos de nuestro grupo de trabajo pero también de las tareas concretas, y de las personas y los proyectos, etc.

## 5.- Ordenación y selección de registros

La ordenación y selección de registros en una base de datos se generan mediante la petición a la aplicación de consultas e informes.

### Añadir tablas y formularios

#### 5.1.- Consultas

Las consultas son las que verdaderamente hacen el trabajo en una base de datos. Pueden realizar numerosas funciones diferentes. Su función más común es recuperar datos específicos de las tablas. Los datos que desea ver suelen estar distribuidos por varias tablas y, gracias a las consultas, puede verlos en una sola hoja de datos. Además, puesto que normalmente no desea ver todos los registros a la vez, las consultas le permiten agregar criterios para "filtrar" los datos hasta obtener solo los registros que desee. Las consultas a menudo sirven de origen de registros para formularios e informes.

Algunas consultas son "actualizables", lo que significa que es posible editar los datos de las tablas base mediante la hoja de datos de la consulta. Si trabaja con una consulta actualizable, recuerde que los cambios se producen también en las tablas, no solo en la hoja de datos de la consulta.

Hay dos tipos básicos de consultas: las de selección y las de acción. Una consulta de selección simplemente recupera los datos y hace que estén disponibles para su uso. Los resultados de la consulta pueden verse en la pantalla, imprimirse o copiarse al portapapeles. O se pueden utilizar como origen de registros para un formulario o un informe.

Una consulta de acción, como su nombre indica, realiza una tarea con los datos. Las consultas de acción pueden servir para crear tablas nuevas, agregar datos a tablas existentes, actualizar datos o eliminar datos.

#### 5.2.- Informes

Los informes sirven para resumir y presentar los datos de las tablas. Normalmente, un informe responde a una pregunta específica, como "¿Cuánto dinero se ha facturado por cliente este año?" o "¿En qué ciudades están nuestros clientes?" Cada informe se puede diseñar para presentar la información de la mejor manera posible.

Un informe se puede ejecutar en cualquier momento y siempre reflejará los datos actualizados de la base de datos. Los informes suelen tener un formato que permita imprimirlos, pero también se pueden consultar en la pantalla, exportar a otro programa o enviar por correo electrónico.

## 6.- Filtros

Se trata de una sofisticada herramienta de selección que permite que se visualicen en la tabla sólo aquellos registros que cumplan con determinada condición (aunque seguirá conteniendo los demás registros, si bien éstos no se mostrarán). Los filtros nos permiten seleccionar aquellos registros que cumplen una propiedad.

Existen básicamente formas diferentes de definir filtros en una tabla (consulta o formulario)

1 .Filtrar por selección, consiste en seleccionar directamente en la hoja de datos (o formulario) la totalidad o parte de un valor y pulsar en el botón Filtro por selección en la barra de herramientas, para que sean seleccionados todos los registros que tengan el valor seleccionado.

2. Filtrar excluyendo la selección, consiste en seleccionar la totalidad o parte de un valor en la hoja de datos (o formulario) y pulsar en el botón Filtro excluyendo la selección en el menú contextual, para localizar todos los registros que no contengan el valor seleccionado.

3. Filtrar por formulario; Cuando abrimos un formulario nos aparece reflejado en él toda la tabla de la base de datos, pero en determinadas ocasiones queremos seleccionar únicamente ciertos registros y no todos, es decir, queremos tener un formulario de una consulta.