

## 5.- TEORÍA SOBRE BASE DE DATOS

---

Antes de proceder al diseño de la base de datos en sí, se va a introducir una descripción general sobre bases de datos.

### **5.1. Tipos de base de datos**

Para un tratamiento “mecanizado” de la información es fundamental que los datos se organicen de forma que se facilite su gestión, es decir, que el acceso a los mismos sea rápido y eficaz.

Convencionalmente, el trabajo en informática de gestión consistía en realizar una serie de programas en determinados lenguajes, para creación, actualización, consulta y listados de datos, que trabajan con información de entrada para obtener unos resultados o datos de salida.

Una base de datos es un conjunto de datos relacionados entre sí, que están estructurados de forma que puede accederse a ellos automática e independientemente de los programas que gestionan esos datos. Esta independencia se refiere a la posibilidad de modificar la estructura de los datos sin necesidad de cambiar los programas que los manipulan, evitando con ello los problemas de actualización de datos previamente existentes.

Toda base de datos está formada por uno o varios bloques de información llamados tablas que normalmente tendrán alguna característica en común.

Una *TABLA* es un conjunto conexo de información del mismo tipo. Cada tabla está formada por registros.

Un *REGISTRO* es la unidad elemental de información de la tabla. Cada registro está formado por uno o más elementos llamados campos.

Un *CAMPO* es cada una de las informaciones que interesa almacenar de cada registro, y es por tanto la unidad elemental de información del registro.

Existen 3 tipos de base de datos que se detallan a continuación.

### 5.1.1. Base de datos jerárquica

Primera en ser utilizada. Se basa en el establecimiento de jerarquías o niveles entre los distintos campos de los registros, basándose en el criterio de que los campos de mayor jerarquía sean los más genéricos, y tiene una estructura arborescente, donde los nodos del mismo nivel corresponden a los campos y cada rama a un registro. En la figura 5.1 se puede observar el concepto de base de datos jerárquicos.

Para acceder a un campo que se encuentra en un determinado nivel, es preciso localizarlo partiendo del nivel superior y descendiendo por las ramas hasta llegar al mismo. Es evidente el problema de lentitud.

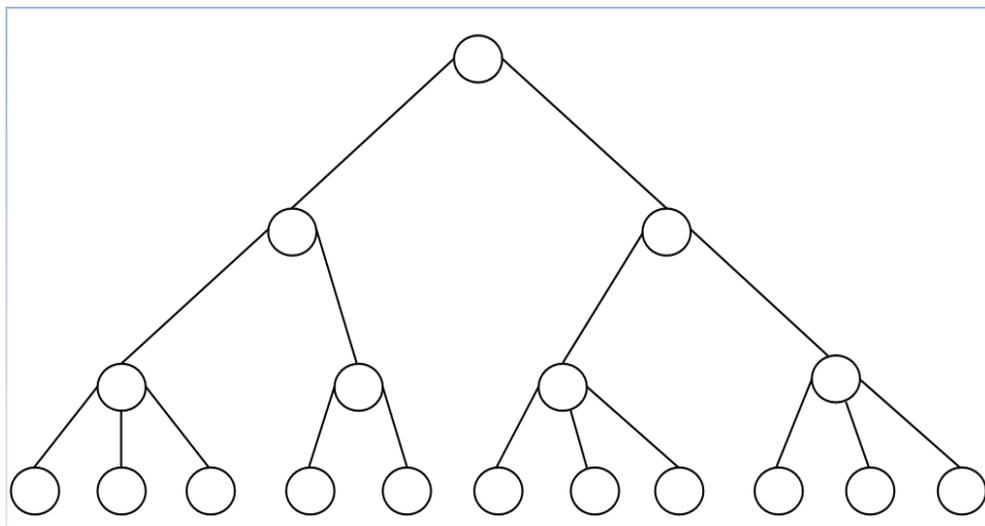


Fig. 5.1- Diagrama base de datos jerárquicos.

### 5.1.2. Base de datos en red

Es una variación del modelo de datos jerárquico. Se utiliza una estructura de grafo, donde existe más de una conexión entre los nodos de diversos niveles, de forma que pueden recorrerse por distintos caminos sin necesidad de acudir cada vez a la primera raíz. Desaparece así el concepto de jerarquía entre campos. Se puede observar este concepto en la figura 5.2.

La principal desventaja es su rigidez. Modificar la estructura de la base de datos requeriría generalmente la reconstrucción de la base de datos completa.

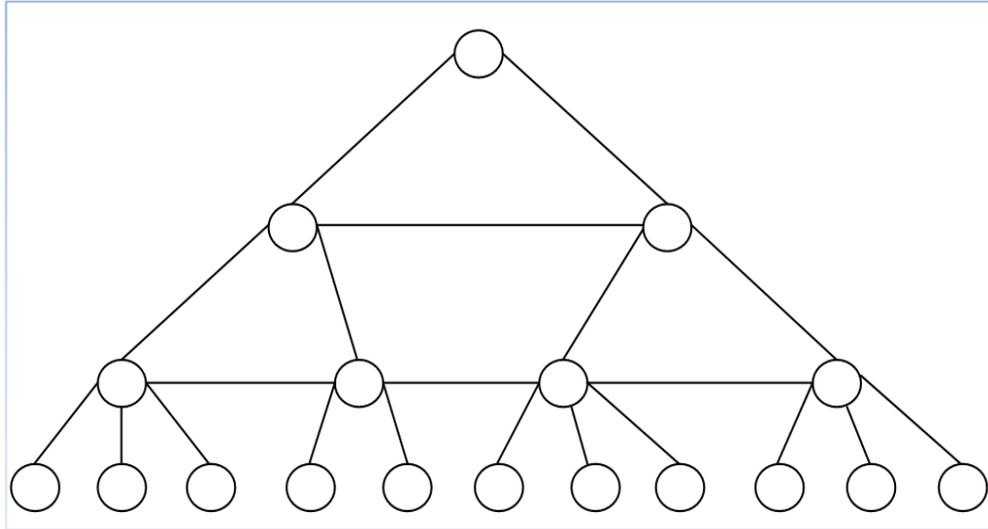


Fig. 5.2- Diagrama base de datos en red

### 5.1.3. Base de datos relacional

El modelo relacional de datos representa todos los datos en estructura de matriz o tabla bidimensional, donde las filas son los registros y las columnas los campos (véase figura 5.3).

Las principales ventajas de la utilización de bases datos relacionales son:

- Actúan sobre las tablas en su conjunto, en lugar de hacerlo sobre los registros.
- Se pueden realizar consultas complejas que utilizan varias tablas de forma simple.
- Son fáciles de utilizar ya que la organización física de los datos es independiente de su tratamiento lógico.

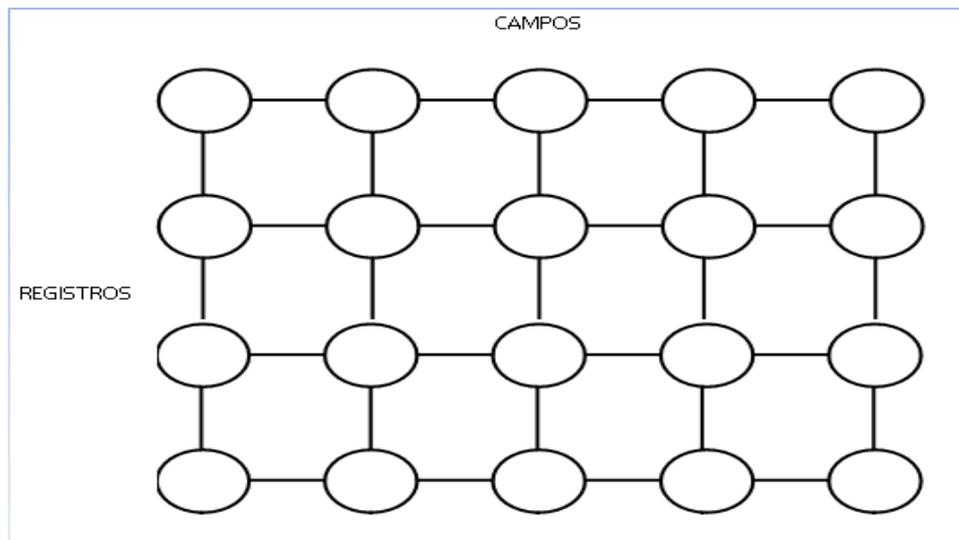


Fig. 5.3- Diagrama base de datos relacional.

## 5.2. Normalización de bases de datos

Son el conjunto de normas que nos ayudan a diseñar una estructura de base de datos óptima para su implementación, gestión y explotación.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

- Cada columna debe tener su nombre único.
- No puede haber dos filas iguales. No se permiten los duplicados.
- Todos los datos en una columna deben ser del mismo tipo.

Las formas normales son aplicadas a las tablas de una base de datos. Decir que una base de datos está en la forma normal N es decir que todas sus tablas están en la forma normal N.

Las formas normales, definidas en la teoría relacional, permiten normalizar una base de datos en un esquema mal diseñado. El creador de las 3 primeras formas normales fue Edgar F. Codd. Cuánto más alta sea la forma normal en la que se encuentran los esquemas de relación, menores serán los problemas que aparecerán en el mantenimiento de la base de datos.

Cuando un esquema de relación está en una forma normal, implícitamente también lo está en las formas normales inferiores a ésta.

### **5.2.1. Primera forma normal (1FN)**

*Se dice que una relación está en 1FN cuando cada atributo sólo toma un valor del dominio subyacente.*

La primera forma normal es inherente al modelo relacional, por lo que su cumplimiento es obligatorio y afecta al número de valores que pueden tomar los atributos de una relación.

### **5.2.2. Segunda forma normal (2FN)**

*Se dice que una relación está en 2FN si está en 1FN y todos los atributos no clave dependen por completo de la clave primaria.*

La segunda forma normal se quebranta cuando la clave de una tabla es compuesta y uno de los campos restantes depende exclusivamente de uno de los campos claves pero no de los otros.

### **5.2.3. Tercera forma normal (3FN)**

*Se dice que una relación está en 3FN si está en 2FN y no existe ningún atributo no principal que dependa transitivamente de alguna clave primaria.*

La tercera forma normal no se cumple cuando existen atributos no principales que dependen funcionalmente de otros atributos no principales.

Éstas son las tres formas normales básicas, aunque existen la forma normal de Boyce-Codd, la cuarta forma normal y la quinta forma normal.

### 5.3. Integridad

Para preservar la consistencia y corrección de los datos almacenados en una base de datos relacional se impone generalmente una o más restricciones de integridad de datos.

Antes de definir las restricciones de integridad se va a detallar una serie de terminología usada en base de datos útil para su comprensión.

➤ *Claves primarias*

Una clave primaria es una columna o combinación de columnas dentro de una tabla cuyo(s) valor(es) identifica(n) unívocamente a cada registro de la tabla.

Se recomienda, aunque no es imprescindible, que todas las tablas tengan alguna columna o grupo de columnas especificadas como clave primaria de la tabla. La clave primaria es un campo (o campos) que nunca puede contener valores NULL.

La clave primaria tiene un valor único diferente para cada fila de la tabla, de modo que no hay dos filas de una tabla con clave primaria que sean duplicados exactos la una de la otra.

Una tabla en donde cada fila es diferente de todas las demás se llama una *relación* en términos matemáticos. El nombre “base de datos relacional” proviene de este término, ya que las relaciones son el corazón de una base de datos relacional.

➤ *Claves externas*

Una clave externa es una columna o combinación de columnas en una tabla cuyo(s) valor(es) es(son) un valor de clave primaria para alguna otra tabla.

Una tabla puede contener más de una clave externa, enlazándola a una o más tablas. Una combinación clave primaria/clave externa crea una relación padre/hijo entre las tablas que las contienen.

Las claves externas son parte fundamental del modelo relacional ya que crean relaciones entre tablas de la base de datos. No tienen sentido si no se van a usar reglas de integridad.

Varios tipos diferentes de restricciones de integridad de datos suelen encontrarse en las bases de datos relacionales, incluyendo:

### **5.3.1.Datos requeridos**

Algunas columnas de una base de datos deben contener un valor de datos válido en cada fila, es decir, no se permite que contengan valores NULL o que falten. Pueden especificarse columnas requeridas cuando se crea una tabla.

### **5.3.2.Chequeo de validez**

Cada columna de una base de datos tiene un dominio, o sea, un conjunto de valores que son legales para esa columna. Pueden especificarse los dominios de las columnas si se precisa.

### **5.3.3.Integridad referencial**

Una clave externa en una base de datos relacional enlaza cada fila de la tabla hijo que contiene la clave externa con la fila de la tabla padre que contiene el valor de clave primaria correspondiente. Se puede forzar esta restricción de clave externa/primaria de modo que se asegura que las relaciones entre entidades de la base de datos se preserven durante las actualizaciones.

### **5.3.4.Integridad de entidad**

La clave primaria de una tabla debe contener un valor único en cada fila, diferente de los valores de todas las filas restantes.

Estos mecanismos de integridad previenen los errores de integridad que se suelen darse tras:

- La inserción de un nuevo registro hijo sin que exista el correspondiente en la tabla padre.
- La supresión de un registro padre que provoca que todos los hijos actuales de esa fila no correspondan con ningún valor de clave primaria en la tabla padre.
- La actualización de la clave primaria en una fila padre provocará que todos los hijos actuales de esa fila queden huérfanos, ya que su clave externa no corresponde con ninguna valor de clave primaria.

### **5.4. Entorno de desarrollo integrado**

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de Integrated Development Environment) no es más que una aplicación informática compuesta por un conjunto de herramientas de programación.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador, etc. Si contienen herramientas para construir interfaz gráfica de usuario se les denomina desarrollo rápido de aplicaciones o RAD (acrónimo en inglés de Rapid Application Development). Los IDEs pueden ser aplicaciones por sí solas o ser parte de aplicaciones existentes.

En el mercado se localizan diferentes herramientas de este tipo que son adecuadas para la creación de aplicaciones de bases de datos, siendo las más destacadas: Microsoft Office Access, Microsoft Visual Basic .NET, Microsoft Visual C# .NET, Microsoft Visual FoxPro, Borland Delphi, etc.

A pesar de la variedad de plataformas disponibles, la elección elegida ha sido Microsoft Access dado que ha sido la única exigencia por parte de la empresa MP a la hora de solicitar la aplicación.

Microsoft Access está basada en Windows por lo que deberá ser instalada en equipos con Microsoft Windows como sistema operativo.

Se va a utilizar la última versión disponible del software, ACCESS 2007, que es la base de datos del paquete integrado Office lo que la convierte en la herramienta más conocida y utilizada para la creación de aplicaciones de bases de datos.

Constituye una herramienta muy eficaz y ampliamente utilizada que ayuda a los usuarios a compartir información, realizar su seguimiento, presentar informes y realizar los trabajos relativos a una base de datos relacional en un entorno interactivo. Permite al programador llevar a cabo desarrollos rápidos y crear aplicaciones de acuerdo a necesidades específicas.

Access dispone del lenguaje de programación Visual Basic para aplicaciones (VBA o Visual Basic for Applications) que le permite extender su funcionalidad y mejorar su rendimiento. VBA no es más que un descendiente de Visual Basic destinado a la edición del comportamiento de las aplicaciones de Microsoft Office.

Access de la opción de ejecutar acciones de macros, que cubren actividades de administración de datos como apertura de formularios, informes, desplazamiento entre registros, etc. Una macro es una lista de una o más acciones que funcionan juntas para realizar una tarea particular como respuesta a un evento.

Para agregar una acción a una macro, puede elegirse de la lista desplegable. Una vez seleccionada, en el panel inferior se despliegan los argumentos relacionados, alguno de los cuales son obligatorios y otros opcionales dependientes de la acción.

## 5.- Teoría sobre base de datos.

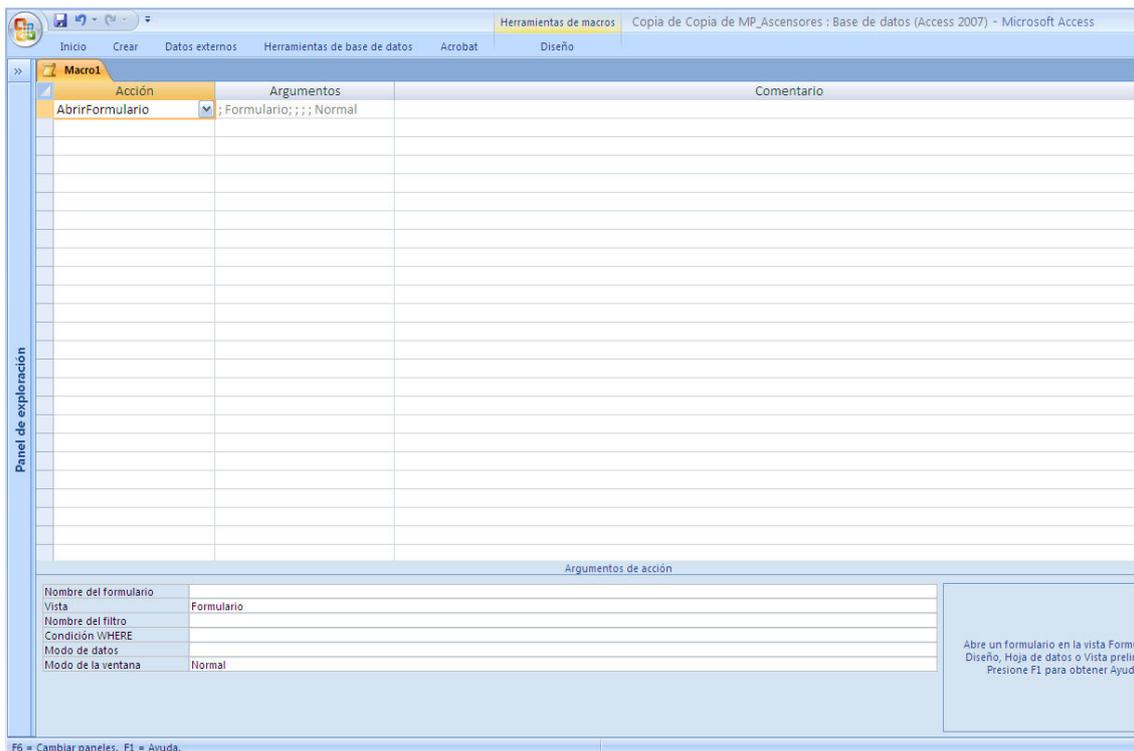


Fig. 5.4- Ventana de macro de Access 2007.

SQL (Structured Query Language) es el lenguaje que Access usa tras bambalinas para programar operaciones de consulta. Está compuesto por instrucciones, que deben cumplir con la sintaxis y las convenciones específicas del lenguaje.

La instrucción **SELECT** es la más común en SQL y la más importante. Todas las consultas de selección empiezan con la instrucción **SELECT**.

Su sintaxis es la siguiente:

```
SELECT [predicado] {* | tabla.* | [tabla.]campo1 [AS alias1],  
          [tabla.]campo2 [AS alias2], [...] ....}  
FROM expresiónTabla[,...]  
[WHERE criterios]  
[GROUP BY ListaCamposGrupo]  
[HAVING criteriosGrupo]  
[ORDEN BY campo1 [ASC | DES] [campo2 [ASC | DES]]...]
```

La tabla 5.1 ilustra las cláusulas de la sentencia **SELECT**.

<b>CLÁUSULA</b>	<b>APARTADO</b>	<b>DESCRIPCIÓN</b>
SELECT	predicado	Uno de los siguientes predicados: ALL, DISTINCT, DISTINCTROW o TOP para limitar el número de registros devueltos. El valor predeterminado es NULL.
	*	Se seleccionan todos los campos de la tabla o tablas especificadas.
	tabla	El nombre de la tabla que contiene los campos de la que se van a seleccionar los registros.
	campo1, campo2	Nombre de los campos que contienen los datos que se desea recuperar.
	AS alias1, AS alias2	Los nombres que se van a usar como encabezados de columnas en vez de los nombres de columnas originales en la tabla.
FROM	expresiónTabla	Expresión que identifica a una o más tablas de las que se van a recuperar datos. La expresión puede ser un nombre de tabla único, un nombre de una consulta guardada o una composición resultante de una cláusula INNER JOIN, LEFT JOIN o RIGHT JOIN.

WHERE	critérios	Expresiones que deben cumplir los registros a obtener. Se usan hasta 40 expresiones vinculadas por operadores lógicos (And, Or y NOT), cláusula IN de pertenencia a un conjunto, una cláusula BETWEEN...AND de contraste de rango y cláusula IN NULL.
GROUP BY	ListaCamposGrupo	Nombres de campos que se van a usar para agrupar registros.
HAVING	critériosGrupo	Expresión que determina qué registros agrupados se van a mostrar.
ORDER BY	Campos[ASC   DES]	Campos por los que se va a ordenar (ascendente o descendente).

Tabla 5.1- Cláusulas de sentencia SELECT.

Para poder ejecutar una aplicación desarrollada con Access el usuario de la aplicación tendrá que tener instalado Access en su equipo. No obstante, existe la posibilidad de generar un programa de instalación personalizado con todas las funciones necesarias para instalar una aplicación desarrollada con Access. Además de instalar los archivos de la aplicación en los sistemas de los usuarios, el programa de instalación puede instalar los archivos run-time de Access (la versión run-time de Access permite ejecutar aplicaciones, pero no cambiarlas o crearlas), crear accesos directos de Windows y configurar los sistemas de los usuarios de muchas otras formas.

Una vez explicado la teoría básica necesaria para poder acometer el diseño de la base de datos, se procederá a pasar al diseño de la base de datos necesaria para el desarrollo de la aplicación.