Tema 1: Almacenamiento de la Información
Concepto de información

Organización lógica de los datos

Una estructura de datos es una colección de datos que se caracterizan por su organización y por las operaciones que sobre dicha estructura puedan definirse. En función de esto, **los datos pueden ser:**

−**Simples**: no se pueden dividir en elementos de tamaño menor. Ejemplo: 1999, como valor de 'Año'.

−**Compuestos**: formados como agregación de datos simples. Ejemplo: 7/Oct/2010, como valor de 'Fecha', compuesto por el d.a (7), el mes (Oct) y el a.o (2o1o).

Un registro.- es la estructura capaz de almacenar datos en forma de unidades homogéneas de información dentro de un almacén de información, es decir, un archivo. Está compuesto por estructuras de menor tamaño denominadas campos. Se distinguen:

Registro lógico: Unidad de información homogénea compuesta por datos referentes a un determinado objeto o concepto, siendo el campo la unidad elemental de información dentro de un registro lógico.

Registro físico: Unidad de transmisión o de almacenamiento de información. Conjunto de información que, en función de las características de la máquina, puede grabarse o leerse de una sola vez. Por lo general, un registro físico está formado por varios registros lógicos

Almacenamiento de la Información

• **Almacenamiento Primario**: Medios sobre los que la CPU puede acceder directamente y por tanto más rápidamente. Es la memoria RAM.

• **Almacenamiento Secundario:** Dispositivos de almacenamiento secundario. En los discos duros cada pista se subdivide en bloques o sectores de tamaño fijo determinado por el S.O (512, 1o24 . 4o96 bytes). La transferencia de información entre memoria y disco tiene lugar en unidades de bloque. Cuando se produce una orden de lectura se copian uno o varios bloques en el llamado buffer de memoria (.rea reservada de Memoria principal), si se realiza una
escritura se copia el bloque correspondiente al disco.

- **Almacenamiento Intermedio:** Es parte de la memoria principal que se utiliza para agilizar la E/S de datos, es el Buffer.

1.- Ficheros

Son estructuras de información que crean los Sistemas Operativos de los ordenadores para poder almacenar datos. Suelen tener un nombre y una extensión que determina el formato de la información que contiene.
El formato y el tipo de fichero determinan la forma de interpretar la información que contiene, realmente el fichero sólo contiene un conjunto binario de o y 1s que debe ser interpretado.

Un archivo, desde el punto de vista lógico, es un conjunto de registros afines considerados, a efectos de un proceso, como una colección de unidades simples de información, de las mismas características en cuanto a estructura, significado y tipo de tratamiento.
Físicamente, se puede definir como un conjunto de datos que tienen entre sí una relación lógica y están almacenados en un soporte de información adecuado para la comunicación con el ordenador. Un archivo almacena información referente a un mismo tema de forma estructurada, con el fin de manipular los elementos que lo componen de forma individual.

Clasificación

Los ficheros pueden ser clasificados de muchas formas:

Según su Contenido, pueden ser:

1.- **Binarios**, tienen que ser interpretados.
- – De imagen: .jpg, .gif, .tiff, .bmp, .wmf, .png, .pcx
- – De video: .mpg, .mov, .avi, .qt
- – Comprimidos: .zip, .rar, .gz, .tar, .Z, .mbz
- – Ejecutables o Compilados: .exe, .a, .o, .cgi, .com
- – Procesadores de Texto: .doc, .docx, .odt

Generalmente los ficheros que componen una BD son binarios. Por ejemplo el software de gestión de la BD Oracle guarda la información en múltiples ficheros (datafiles, tempfiles, logfiles, etc.), Access guarda la información de una BD con extensión .mdb o .accdb

2.- **De Texto:** También llamados Ficheros planos o ficheros ASCII (American Standard Code for Información Interchange) Tablas ascii/unicode. Los ficheros de texto, no necesitan ser interpretados, pero suelen tener extensiones que sirven para conocer el tipo de información que contienen. Por ejemplo:
- - Ficheros de configuración: .ini, .inf, .conf
- - Ficheros de c.digo fuente: .java, .c, .sql
- - Ficheros de P.ginas web: .html, .php, .css, .xml
- - Formatos enriquecidos: .rtf, .ps, .tex

Según su Organización:

En función de la organización interna de los datos:
1.- **Secuencial**: Los registros se escriben sobre el dispositivo de almacenamiento en posiciones físicamente contiguas.
Ventajas: velocidad de acceso, fragmentación extema mínima.
Inconvenientes: consulta secuencial, dificultad en la inserción o eliminación de registros intermedios (huecos).

Los archivos secuenciales son típicamente utilizados en aplicaciones de proceso de lotes y son óptimos para dichas aplicaciones si se procesan todos los registros. La organización secuencias de archivos es la única que es fácil de usar tanto en disco como en cinta. En el caso de registros de longitud fija, el acceso al registro i vendrá dado por: N.Bloque=i/FB y el registro concreto estar. en i mod FB. Para las aplicaciones interactivas que incluyen peticiones o actualizaciones de registros individuales, los archivos secuenciales ofrecen un rendimiento pobre. Hay variantes:

− **Secuencial encadenada.** Los registros contienen un campo adicional que indica qué registro es el siguiente en la secuencia lógica.
Ventajas: velocidad de acceso, facilita la inserción de nuevos registros.
Inconvenientes: aumento de tamaño para almacenar la posición del siguiente, la eliminación es lógica (no libera espacio físico).

− **Secuencial indexada:** Se cede un espacio del archivo para almacenar las direcciones de los registros, a los que se accede secuencialmente.

Ventajas: velocidad de acceso, facilita la inserción de nuevos registros.
Inconvenientes: aumento de tamaño para almacenar las posiciones, aumento de tiempo en las operaciones de actualización.

Los registros se organizan en una secuencia basada en un campo clave presentando dos características, un **índice** del archivo para soportar los accesos aleatorios y un **archivo de desbordamiento**. El índice proporciona una capacidad de búsqueda para llegar rápidamente al registro deseado y el archivo de desbordamiento es similar al archivo de registros usado en un archivo secuencial, pero está integrado de forma que los archivos de desbordamiento se ubiquen siguiendo un puntero desde su registro predecesor.
Cada registro del archivo índice tiene dos campos, un campo clave igual al del archivo principal y un puntero al archivo principal. Para encontrar un campo específico se busca en el índice hasta encontrar el valor mayor de la clave que es igual o precede al valor deseado de la clave, la búsqueda continua en el archivo principal a partir de la posición que indique el puntero.

Las características más relevantes de un archivo indexado, son las siguientes:

- Permiten utilizar el modo de acceso secuencial y el modo de acceso directo para leer la información guardada en sus registros.
- Solamente se puede grabar en un soporte direccionable.
- El diseño del registro tiene que tener un campo, o combinación de campos, que permita identificar cada registro de forma única, es decir,que no pueda haber dos registros que tengan la misma información en él.

Organización Directa:

Se utiliza una clave numérica que identifica los registros que componen un archivo. Normalmente se utilizan funciones de Hash para transformar la clave en una posición física.

Ventajas: acceso inmediato a los registros a partir de su clave, permite realizar operaciones simultáneas de lectura/escritura, muy rápidos en el tratamiento individual de registros.
Inconvenientes: consultas lentas para el fichero completo, el disco puede contener huecos libres. Sólo se puede utilizar con registros de longitud fija.

En función del uso que se hace de ellos:
Permanentes: se almacenan en memoria auxiliar, para permitir su uso posterior aunque se apague el equipo. A su vez, se clasifican en:
- Maestros.
- Constantes.
- Históricos

Temporales: su tiempo de vida es de corta duración

Métodos de acceso

Para poder utilizar la información almacenada en un archivo, las aplicaciones deben acceder a la misma y almacenarla en memoria. Hay distintas formas de acceder a un archivo. Para una aplicación, elegir adecuadamente la forma de acceder a un archivo suele ser un aspecto importante de diseño, ya que, en muchos casos, el m.todo de acceso tiene un impacto significativo en el rendimiento de la misma.

Dependiendo de que se pueda ir de una posición a otra de un archivo, pasando por todas las posiciones anteriores o no, se distinguen dos métodos de acceso principales: **acceso secuencial** y **acceso directo.**

Procedimiento seguido para acceder a uno o más registros determinados de un fichero.

Se utilizan índices para hacer más efectiva la búsqueda. Los índices son estructuras de datos que relacionan valores de un campo (normalmente el campo clave) de un registro con su dirección de memoria. Una primera c**lasificación de índices** es la siguiente:

- Índice Denso: si contiene una referencia a todos los registros del fichero.
- Índice no Denso: si sólo contiene referencia a un subconjunto de registros. Se suelen definir cuando el orden del fichero y del índice coincide.

Otra clasificación de Tipos de índices es:
– Índices primarios: el índice es el campo clave de ordenación del fichero (su valor no puede repetirse en otro registro del fichero).
– Índices de agrupamiento: el campo de ordenación no es el campo clave, es decir, pueden haber valores repetidos .
– Índice secundario: cuando el campo de indización no es el de ordenación.

Inconvenientes de los sistemas de ficheros:

Separación y aislamiento de los datos.
Duplicación o Redundancia de datos que origina fallos en la Integridad de la información.
Dependencia de la codificación de los datos en los programas.
Formatos de ficheros incompatibles.

Estos inconvenientes de los sistemas de ficheros se pueden atribuir a dos factores:

1.- La definición de los datos se encuentra codificada dentro de los programas de aplicación, en lugar de estar almacenada por separado de forma independiente.

2.- No hay control sobre el acceso y la manipulación de los datos, fuera de lo impuesto por los programas de aplicación.

2.- Evolución de los sistemas de ficheros a las bases de datos

En los sistemas de información clásicos soportados sobre ficheros los datos se recogen varias veces y se encuentran repetidos en distintos archivos. Esta redundancia, además de malgastar recursos, puede producir divergencias en los resultados. Estos sistemas, ponen el énfasis en el tratamiento que reciben los datos, que se organizan en ficheros que se diseñan para una determinada aplicación.

Las aplicaciones se diseñan e implantan con total independencia unas de otras y los datos no se suelen transferir entre ellas, sino que se duplican siempre que los distintos trabajos los necesitan.

El sistema clásico de gestión de ficheros genera además una ocupación excesiva de memoria secundaria, un aumento de los tiempos de proceso, al repetirse los mismos controles y operaciones en los distintos ficheros y, sobre todo, inconsistencias debidas a que la actualización de los mismos datos que, cuando se encuentran en más de un fichero, no puede realizarse de forma simultánea en todos ellos.

La solución es utilizar un “sistema orientado a los datos” en el que los datos sean recogidos y almacenados una sola vez, con independencia de los tratamientos que sobre los datos se realicen.

En una base de datos, los datos se mantienen y organizan en un conjunto estructurado que no está diseñado para una aplicación concreta, sino que, por el contrario, tiende a satisfacer las necesidades de información de toda la organización.

E. F. Codd.- modelo relacional 1970.
Antes punteros físicos (direcciones de disco) relacionaban registros de distintos archivos.
Relacionar registro A con registro B-> añadir a registro A un campo conteniendo la dirección en disco de registro B. Este campo siempre señalaría desde el registe A al registro B.

Vulnerabilidad: añadir un nuevo disco y mover los datos de una localización física a otra: convergión de los archivos de datos.

Sistemas basados en modelo de red y modelo jerárquico, primera generación de los SGBD.

Sistemas de bases de datos
Una base de datos es un conjunto de datos entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

Problemas de sistemas de ficheros:
- separación/aislamiento de los datos
- duplicación de datos
- dependencia
- formato
- concurrencia
- autorizaciones
- catálogo

Niveles de abstracción en una base de datos:

1.- **Conceptual** : vista del usuario. Es el que percibe el usuario a través de las aplicaciones.

2.- **Lógico** : representación global de los datos, independiente tanto del hardware como de cada usuario en particular. Debe incluir:

−La descripción de todos los datos.
−La descripción de las interrelaciones entre los datos.
−Las restricciones de integridad y de confidencialidad

3.- **Físico**: datos almacenados en forma de bits. Debe especificar, al menos:
– Estrategia de almacenamiento.
– M.todos y rutas de acceso.
– Otros: compresión, encriptado, ajustes, etc

3.- Concepto de Bases de Datos

Es una colección de información perteneciente a un mismo contexto, que está almacenada de forma organizada en ficheros en soporte secundario (no volátil) y con redundancia controlada. Los datos deben ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se debe apoyar en un modelo de datos, que permite captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitar.n la seguridad del conjunto de los datos.

En una base de datos debe existir independencia entre los programas y los datos que los manejan. Por otro lado los mismos datos pueden ser utilizados por distintas aplicaciones y usuarios.

La base de datos debe permitir métodos para **consultar, insertar, modificar y eliminar** datos.

Una base de datos almacena información de dos tipos:

1. Datos del usuario, el contenido de la base de datos.
2. Datos del sistema, tanto la estructura como los datos de control necesarios para su manejo.

El Sistema de Gestión de Bases de Datos es el conjunto de programas que permiten la implantación, acceso y mantenimiento de una base de datos. **El SGBD y la base de datos constituyen el Sistema de Base de Datos.**

Componentes de una Base de Datos

- **Dato**: Porción de información concreta sobre un concepto o suceso.
- **Tipo de dato**: indica la naturaleza del campo (numérico, alfanumérico, compuesto)
- **Campo o Columna**: es un identificador para un dato. Cada campo pertenece a un tipo de dato.
- **Registro, Tupla o Fila**: Conjunto de datos referente a un mismo concepto o suceso.

- **Campo Clave**: Campo especial que identifica de forma única a cada registro.
- **Tabla**: Conjunto de registros bajo un mismo nombre que representa al conjunto de todos ellos.
- **Consulta**: Instrucción para hacer peticiones a una BD. Puede ser una consulta simple de un registro concreto o una solicitud para seleccionar todos los registros que satisfagan un conjunto de criterios. Así como una petición de inserción, eliminación o modificación/actualización de registros.
- **Índice**: Estructura que almacena los campos clave de una tabla, organizándolos para hacer más fácil encontrar y ordenar los registros de la tabla. Para buscar un elemento que está indexado, sólo hay que buscar en el índice de dicho elemento para, una vez encontrado, devolver el registro que se encuentre en la posición marcada por el índice.
- **Vista**: Transformación que se hace a una o más tablas para obtener una nueva tabla virtual, es decir, que no está almacenada en memoria secundaria, aunque si se almacena su definición.
- **Informe**: Listado ordenado de los campos y registros seleccionados.
- **Guiones o Scripts**: Conjunto de instrucciones, que ejecutadas de forma ordenada, realizan operaciones de mantenimiento de datos en la BD.
- **Procedimientos**: Tipo especial de Scripts que está almacenado en la BD y forma parte de su esquema.

Los datos se almacenan a través de un **Esquema**, que es la definición de la estructura donde se almacenan los datos, contiene todo lo necesario para organizar la información mediante tablas, registros y campos. También contiene otros objetos necesarios para el tratamiento de los datos (procedimientos, vistas, índices, etc.). También se le llama Metainformación, es decir, información sobre la información o metadatos. Los gestores suelen almacenar el esquema en tablas.

Evolución y Tipos BD

Según ha ido avanzando la tecnología, las BD han ido cambiando la forma de representar y extraer la información. Década de 1950 unida a la utilizaci.n de cintas magn.ticas => aplicaciones basadas en Sistemas de Ficheros.

Década de 1960 se generaliza la utilización de los discos magnéticos => BD Jerárquicas y en Red. El modelo jerárquico se parece a un árbol invertido: Representa bien las relaciones de 1 a N, pero es más costoso representar otro tipo de relaciones. Además resulta complicado añadir nuevas relaciones y el Desarrollo de aplicaciones resulta complejo ya que El programador necesita conocer la estructura de datos utilizada.

El modelo en Red se desarrolló a partir del modelo jerárquico para resolver su falta de flexibilidad, por ejemplo, permite que un nodo de nivel inferior (hijo o miembro) tenga dos nodos de nivel superior(padre o propietario), permitiendo modelar relaciones de N a M.
Este modelo resulta mucho más difícil de implementar y mantener que el jerárquico y sigue presentando problemas con las relaciones, además el programador sigue teniendo que conocer bien las estructuras de datos para lograr que el modelo sea eficiente.

Década de 1970 Edgar Frank Codd publica el Modelo Relacional de Datos, basado en la lógica de predicados y teoría de conjuntos.

• Este Modelo supuso un gran avance, en vez de basarse en una Relación de superior a inferior, permite que cualquier fichero se relacione con otro a través de un campo común.

La complejidad se redujo mucho, ya que los cambios se podían realizar sobre el esquema de la BD sin que ello afectara a la capacidad del sistema para acceder a los datos. Pero este modelo resultó inaplicable en 1970, ya que la facilidad de uso presentaba un coste importante de rendimiento y el hardware en aquellos a.os no podía implementarlo.

Década de 1980 IBM lanza su motor de BD DB2 para la plataforma MVS. Y despu.s IBM crea el SQL.
Década de 1990 IBM lanza su nueva versi.n de DB2 para BD paralelas.
Tras la aparición de Intemet y el crecimiento de la información, se crean las BD Distribuidas, se multiplica el número de ordenadores que controlan una BD, surgiendo las BD Multidimensionales, que forman los denominados 'cubos de información'.

Bases de datos relacionales

En el modelo relacional, la información se representa en forma de tablas, que almacenan información lógicamente relacionada. Estas tablas se relacionan formando vínculos o relaciones entre ellas que se establecen implícitamente por la presencia de campos de conexión comunes.

Cada tabla se compone de filas y columnas, cada fila almacena un registro con tantos campos como columnas tenga la tabla.

Bases de Datos orientadas a objetos

Las bases de datos orientadas a objetos surgen para cubrir nuevos tipos de aplicaciones:
−Diseño y fabricación asistido por ordenador (CASE,CAD/CAM).
−Bases de datos gráficas y de im.genes.
−Bases de datos científicas.
−Sistemas de información geográfica.
−Bases de datos multimedia.
−Acceso uniforme a sistemas de múltiples bases de datos.
−Gestionan objetos en los cuales están encapsulados los datos y las operaciones que actúan sobre ellos.
−Desarrollo bajo orientación a objetos un único modelo subyacente implementado en el sistema de datos, al que pueden acceder directamente las aplicaciones.
−Intentan satisfacer necesidades de aplicaciones más complejas que las tradicionales que utilizan nuevos tipos de datos para almacenar im.genes o grandes bloques de texto.
−Permiten al diseñador de la base de datos especificar la estructura de objetos complejos as. como las operaciones que se pueden aplicar a estos objetos

Bases de datos documentales

Una base de datos se crea y mantiene de forma continuada con el objetivo de resolver necesidades de información concretas de un colectivo, una empresa o el conjunto de la sociedad. Las bases de datos documentales pretenden cubrir las nuevas necesidades de información.

En una base de datos documental, cada registro se corresponde con un documento, de tipo:
− Audiovisual, gr.fico o sonoro.

− Publicación impresa.
− Electrónico.
− De archivo.
− Etc.

Estos recursos electrónicos pueden consultarse directamente en formato electrónico o utilizarse para elaborar productos impresos, como bibliografías, informes, etc.

3.4.3.- Bases de datos documentales

• En función del propósito, se distinguen:
−Cientifico-tecnológicas
Multidisciplinares: abarcan varias disciplinas científicas o técnicas.
Especializadas.
−Económico-empresariales.
−De medios de comunicación.
−Del ámbito político-administrativo y jurídico.
−Del ámbito sanitario.
−De información general

Usos de las BD

• Los usos más frecuentes de las BD son:
BD Administrativas
BD Contables
BD para Motores de Búsqueda
Científicas
Configuraciones
Bibliotecas
Censos
Virus
Militares
VideoJuegos
Deportes

Criterios de uso de ficheros o bases de datos

El uso de un sistema de ficheros:
− Facilita la localización de la información en el dispositivo de almacenamiento secundario.
− Reduce el tiempo de desarrollo.
− Asegura un control exhaustivo sobre el contenido de los archivos.
− Reduce el tiempo de respuesta con archivos de gran tamaño.
− Permite adaptar el formato de modo que el usuario pueda consultar su contenido

El uso de una base de datos aporta:
- Mayor eficiencia en la recogida, validación e introducción de los datos en el sistema
- Reducción del espacio de almacenamiento.
- Independencia de los datos respecto de los programas que los utilizan y viceversa
- Asegura la coherencia de los resultados.
- Mejor disponibilidad de los datos para el conjunto de los usuarios.

- Mayor calidad y normalización en la documentación de la información, que se integra con los dato

5.- Sistemas Gestores de Bases de Datos (SGBD)

Un SGBD es el conjunto de herramientas que facilitan la consulta, uso y actualización de una BD. Un ejemplo es Oracle 11g, que incorpora un conjunto de herramientas software que son capaces de estructurar en múltiples discos duros los ficheros de una BD, permitiendo el acceso a sus datos tanto a partir de herramientas gráficas como a partir de potentes lenguajes de programaci.n (PL-SQL, php, c++ …)

Ejemplos de gestores de base de datos libres:
Firebird
BDB
MySQL
PostgreSQL
Sqlite

Ejemplos de gestores de base de datos propietarios:
dBase
FileMaker
Fox Pro
IBM DB2 Universal Database (DB2 UDB)
IBM Informix
MAGIC
Microsoft SQL Server
Open Access
Oracle

5.- Concepto de SGBD

Funciones de un SGBD (1)

1.- Permiten a los usuarios almacenar datos, acceder a ellos y actualizarlos de forma sencilla

2.- Garantizan la integridad de los datos, no permitiendo operaciones que dejen un conjunto de datos incompleto o incorrecto.

3.- Proporcionan, junto al S.O, un sistema de seguridad que garantiza el acceso a la información exclusivamente a los usuarios autorizados.

4.- Proporcionan un diccionario de metadatos, que contiene el esquema de la BD y que es fácilmente accesible.

5.- Permite el uso de transacciones, garantizando que todas las operaciones de la transacción se realicen correctamente, y en el caso de que haya alguna incidencia se deshagan los cambios.

6.- Ofrecen estadísticas sobre el uso del gestor, posibilitando la monitorización del uso de la BD.

7.- Permiten la concurrencia.

8.- Independizan los datos de la aplicación o usuario que está utilizándolos, haciendo más f.cil su migración a otras plataformas.

9.- Ofrecen conectividad con el exterior. El protocolo ODBC (Open DataBase Connectivity) se utiliza frecuentemente como forma de comunicación entre BD y aplicaciones extemas.

1o.- Incorporan herramientas para realizar copias de seguridad y restauración de la información.

Lenguaje SQL(Structured Query Language)

Es el interfaz de comunicación con el usuario. Está estandarizado por la ISO (Intemational Organización for Standarization). Se compone de cuatro lenguajes:

DDL: Lenguaje de Definición de Datos. Se ocupa de la creaci.n de la estructura de la BD (CREATE, DROP)
DML: Lenguaje de Manipulación de Datos. Permite seleccionar datos (SELECT), insertarlos (INSERT), eliminarlos (DELETE) y actualizarlos (UPDATE).
DCL: Lenguaje de Control de Datos. Incluye GRANT y REVOKE, que permiten al administrador gestionar el acceso a los datos contenidos en la BD.
TCL: Lenguaje de Control de Transacciones. Permite ejecutar varios comandos de forma simultánea como si fuera un comando atómico o indivisible (COMMIT) y si ocurriera algún incidente podría deshacer los pasos dados (ROLLBACK).

El Lenguaje de definición de datos (DDL, Data Definition Language) Permite especificar los datos, su estructura y las relaciones que existen entre ellos. Debe proporcionar los datos para los niveles intemo, conceptual y extemo, especificando las características de los mismos en cada nivel:

− A nivel intemo, se debe indicar el espacio Zsico para los datos, tamaño y Spo de datos, as. como los m.todos de acceso.

− A nivel conceptual y extemo, se deben definir enSdades, sus atributos, interrelaciones entre ellos, permisos de acceso, etc.

El lenguaje de manipulación de datos (DML, Data Manipulation Language) Permite recuperar, añadir, suprimir o modificar los datos respetando siempre las normas de seguridad y especificaciones del administrador. Proporciona un conjunto de procedimientos que facilitan las tareas del administrador.

• El Lenguaje de Control de Datos (LCD o DCL, Data Control Language). Permite:
− Gestionar la capacidad de los ficheros.
− Obtener estadísticas de uso.
− Gestionar los procedimientos de copia de seguridad.
− Rearranque en caso de caída del sistema.
− Protección frente a accesos no autorizados.

5.4.- Tipos

Según su capacidad y potencia se pueden distinguir Los GBD Ofimáticas, como el Microsoft Access con VBA (Visual Basic for Applications) como lenguaje y los GBD Corporativos como Oracle, DB2 y MySQL.

6.- Usuarios de la base de datos

ANEXO I: ALGORITMOS DE Búsqueda
Búsqueda secuencial o lineal. Este m.todo de búsqueda es muy lento, pero si los datos no están en orden es el único m.todo que puede emplearse para hacer las búsquedas. Si los valores de la clave no son únicos, para encontrar todos los registros con una clave particular, se requiere buscar en toda la lista.

Consiste en recorrer la lista elemento a elemento e ir comparando con el valor buscado (clave). Se empieza con la primera posición del array y se observa una casilla tras otra hasta que se encuentra el elemento buscado o se han visto todas las casillas.
El número máximo de accesos (comparaciones) para este algoritmo es de n. El m.nimo de 1 y en promedio n/2.

ANEXO I: ALGORITMOS DE Búsqueda
Búsqueda Binaria o Dicotómica
• Los prerrequisitos principales para la búsqueda binaria son:
La lista debe estar ordenada en un orden específico de acuerdo al valor de la clave.
Debe conocerse el número de registros.
• La búsqueda binaria utiliza un m.todo de `divide y vencerás‘ para localizar el valor deseado. Con este método se examina primero el elemento central de la lista; si éste es el elemento buscado, entonces la búsqueda ha terminado.
• En caso contrario, se determinar si el elemento buscado ser. en la primera o la segunda mitad de la lista y a continuación se repite este proceso, utilizando el elemento central de esa sublista.
• El número máximo de accesos para este algoritmo es de log2n. El mínimo de 1 y en promedio . log2 n.

ANEXO I: ALGORITMOS DE Búsqueda
Búsqueda Binaria o Dicotómica

```
elem=29
lInf=o
lSup=19
mitad=(LInf+LSup)/2
encontrado=false
MIENTRAS (lInf <= lsup AND encontrado==false)
Si(elem>Datos[mitad])
lSup=mitad+1
Si no
Si(elem>Datos[mitad])
lInf=mitad-1
Si no
encontrado=false
FIN MIENTRAS
Si (encontrado)
Visualizar “Elemento encontrado”
Si no
Visualizar “Elemento no encontrado”
```

Una búsqueda binaria de, por ejemplo, el 29 requerir.a 4 lecturas mientras que una secuencial requerir.a 11 .

Tema 2: Interpretación de Diagramas Entidad / Relación

1

Modelo Conceptual.

1.- Introducción

Una BD representa la información contenida en algún dominio del mundo real. El diseño de una BD consiste en extraer todos los datos importantes de un problema. Para obtener estos datos se tiene que hacer un análisis en profundidad del problema, de ese análisis surge el documento E.R.S. (Especificación de Requisitos Software) que extrae toda la información necesaria para la modelización de Datos.

2.- El Modelo de Datos

La modelización consiste en representar el problema realizando múltiples abstracciones, para reunir toda la información sobre un problema y generar un mapa donde están identificados todos los objetos de la BD. En este proceso se suelen utilizar tres Modelados:

- El Modelo Conceptual. Trata de representar el problema tal y como el Usuario lo concibe. Normalmente se utiliza el Modelo Entidad/Relación (MER).
- El Modelo Lógico. Trata de representar el problema de una forma más técnica. El modelo lógico elegido depender. de la implementación de la BD. Normalmente se utiliza el Modelo Relacional si se trabaja con una BD Relacional.
- El Modelo Físico. Es el resultado de aplicar el Modelo Lógico a un SGBD concreto. Será la transformación del Modelo Relacional en modelo físico a través del sublenguaje DDL de SQL.

3.- Modelo Entidad/Relación

Fue desarrollado por Peter Chen en 1976. Es un método de representación abstracta del mundo real centrado en las restricciones o propiedades lógicas de una BD y que precede al Modelo Relacional. No es directamente aplicable en un SGBD, sino que necesita una transformación a las estructuras de datos del modelo de datos propio del SGBD empleado. Es un diagrama inicial en el proceso de dise.o que sigue varias etapas hasta llegar a un modelo físico final codificado en el lenguaje mediante el DDL de SQL.

3.- Diagramas E/R

• Para representar el modelo conceptual se usa el modelo Entidad/Relación que es una técnica de representación gráfica que incorpora información relativa a los datos y la Relación existente entre ellos.

• El modelo entidad-Relación es un paso previo al futuro dise.o de una base de datos y, por tanto, independiente del Sistema Gestor de Bases de Datos que se utilice para su implantación física.

3.1.- Entidad

Una entidad es cualquier tipo de objeto o concepto que existe, que puede distinguirse de otros y del cual se desea almacenar información. Se representa mediante un rectángulo.
• Se denomina entidad a la estructura genérica.
• Se denomina ocurrencia de entidad a cada una de las realizaciones concretas de la misma. Es una instancia de una determinada entidad.
Ejemplo:
–Entidad: PERSONA
–Ocurrencia de entidad: Pepe Rodriguez con D.N.I 22116323H y fecha de nacimiento 10 de Junio de 1990.

Tipos de entidades

• **Entidades fuertes, propias o regulares**: Son aquellas cuyas ocurrencias son identificables por sí mismas. Los atributos que las identifican son propios de la entidad. Ejemplo ALUMNO ENTIDAD FUERTE identificable por su n. de matrícula. Se representan mediante un rectángulo simple.

• **Entidades débiles**: Aquellas cuyas ocurrencias son identificables solamente por estar asociadas a otra entidad, es decir, alguno de los atributos que las identifican se refiere a otra entidad. Su existencia depende de la existencia de otra entidad. Si se elimina una ocurrencia de la entidad fuerte, desaparecen también con ella todas las ocurrencias de la entidad débil dependientes de la misma. Las entidades débiles se representan mediante un rectángulo doble.

3.2.- Atributo

• Un atributo es una unidad básica e indivisible de información acerca de una entidad o una Relación que sirve para identificarla o describirla.
• Se puede definir, también, como cada una de las propiedades o características que tiene una entidad o Relación.
• Ejemplo: la entidad AUTOR tiene como atributos el Nombre, la Nacionalidad, la Fecha de Nacimiento, etc.
• El conjunto de posibles valores que puede tomar un atributo recibe el nombre de dominio. Por ejemplo en el atributo COLOR el dominio podría ser ROJO, BLANCO, AZUL Y NEGRO. La representación gráfica de un atributo es una elipse etiquetada con el nombre del mismo.
• Por su funcionalidad los atributos pueden ser:
– Atributo Identificador Principal o clave Principal (distingue unívocamente una ocurrencia de entidad del resto de ocurrencias, si hay más de uno se llamar.n AICandidatos, uno ser. el AIP y los demás AIAlternativos o Secundarios).
– Atributo Descriptor (caracteriza una ocurrencia, pero no la distingue de las demás).

• DOMINIO (values set)

Conjunto de valores. Cada atributo simple está asociado a un dominio, que especifica sus valores válidos.
Atributo Identificador: Conjunto de uno o más campos que identifican unívocamente una ocurrencia de una entidad, por ejemplo, el DNI de un alumno., el teléfono sería un AISecundario o Alternativo.
Atributo opcional/obligatorio.- según si tiene o no necesariamente un valor para todas las ocurrencias de una entidad, por ejemplo el dni de un alumno.
- Opcionales: Pueden tomar valores nulos
- Obligatorios: Deben tener un valor para cada ocurrencia de la entidad.

Según su cardinalidad:
- Multivaluados: Para una misma ocurrencia de entidad pueden tomar varios valores (Una persona puede tener más de un teléfono)
- Univaluados: Un .nico valor para cada ocurrencia.

Según su carácter:
- Simples: Su valor lo introduce el usuario
- Derivados: Campos calculados a partir de datos simples. Por ejemplo saber la mayoría de edad a partir de la fecha de nacimiento.

Representación de Atributos

3.3.- Relación

• Una relación es una asociación, sin existencia propia, de varias entidades. Cada Relación tiene un nombre que distingue su función. Se representa mediante un rombo, generalmente el nombre de una Relación es un verbo, ya que representa acciones entre dos o más entidades.

Clasificaciones de las Relaciones: según su grado o número de entidades que participan en la relación:

Unarias o reflexivas de grado 1
Binarias de grado 2
Ternarias de grado 3
n-arias de grado > 3

• Relaciones Unarias o reflexivas: Ejemplo entre una persona y su cabeza de familia (que tambi.n es persona). Ejemplo: relación entre una persona y su jefe.

Nombres de Rol (papel)
• Todo tipo de entidad que participa en un tipo de Relación juega un papel espec.fico en la Relación. Los nombres de rol se deben usar, sobre todo, en los tipos de Relación reflexivos, para evitar ambigüedad.

• Relaciones n-arias (grado >3) Son aquellas en las que participan más de tres entidades.

3.4.- Participación o Cardinalidad de una Entidad

La cardinalidad de una entidad informa del grado de Participación de dicha entidad concreta en la Relación.
La Participación de una ocurrencia de una entidad, indica mediante una pareja de números, el mínimo y el máximo de veces que puede aparecer en la Relación asociada a otra ocurrencia de entidad. Las posibles participaciones son : (0,1) (1,1) (0,n) (1,n)
Las reglas que definen la Participación de una ocurrencia en una Relación son las reglas de negocio, se reconocen a través de los requisitos del problema.

3.5.- Tipo de Correspondencia de una Relación

• TIPO DE CORRESPONDENCIA: Se calcula a través de las participaciones de sus ocurrencias en ella. Se toma el número máximo de participaciones de cada una de las entidades en la Relación. Por ejemplo, en el caso (2) la cardinalidad sería 1:N, porque por el lado de la Categoría el máximo es 1 y por el del Producto N. 1:N

3.5.1- Clasificación de los tipos de Correspondencia

• 1:1 => Una entidad A puede estar vinculada mediante una Relación a una y sola una entidad B y viceversa. Por ejemplo, se puede limitar el n.mero de Jefes de Departamento mediante una Relación 1:1, as. un Profesor sólo puede ser Jefe de un Departamento, y un Departamento sólo puede tener un Jefe.
• Un Coche tiene una plaza de Garaje y a una plaza de Garaje se le asigna un Coche como máximo. (1,1) (0,1)
EMPLEADO Dirige DEPARTAMENTO

3.5.1- Clasificación de los tipos de Correspondencia

• 1:N => Una entidad A puede estar vinculada mediante una Relación a varias ocurrencias de otra entidad B. Sin embargo, una de las ocurrencias de la entidad B sólo puede estar vinculada a una ocurrencia de la entidad A. Por ejemplo, un Representante gestiona la carrera de varios músicos, sin embargo, un Músico sólo puede tener un Representante.

• Un Coche tiene varias Piezas y cada Pieza está asociada a un solo Coche.
1:N (0,1) (0,n)

Clasificación de Cardinalidades

• M:N o N:M Especifica que una entidad A puede estar vinculada mediante una Relación a varias ocurrencias de otra entidad B, y a su vez, una ocurrencia de la entidad B puede estar vinculada a varias ocurrencias de la entidad A. Por ejemplo, un empleado puede trabajar en varios proyectos y en un proyecto pueden trabajar varios empleados.
N:M
(1,n) (1,n)

OTRAS NOMENCLATURAS: Puntas de flecha: La línea de la Relación que termina en flecha indica la rama N de la cardinalidad de la Relación.
Notación classic de MySQLWorkbench: Las relaciones se expresan con un rombo, rellenando en negro la mitad de la figura en el lado de la entidad cuya cardinalidad es N

3.6.- Cardinalidad de relaciones no binarias

• Para calcular la cardinalidad de una Relación ternaria se tomar. una de las tres entidades y se combinan las otras dos. Después se calcula la Participación de la entidad en la combinación de las otras dos. Posteriormente se hará lo mismo con las otras dos entidades. Finalmente tomando los máximos de las participaciones se generan las cardinalidades.
(0,n) (0,1)

3.7.- Cardinalidad de las relaciones reflexivas

En las relaciones reflexivas, la misma entidad juega dos papeles distintos en la Relación. Para calcular su cardinalidad hay que extraer las participaciones Según los dos roles existentes. Por ejemplo, en la Relación “Es Jefe”, un Empleado tiene el rol de Jefe y el rol de Subordinado. Para calcular las participaciones, hay que contestar a las preguntas:
¿Cuántos Subordinados puede tener un Jefe? (1,n)
¿Cuántos Jefes puede tener un Subordinado? (0*,1)
*Teniendo en cuenta que puede existir un Subordinado sin Jefe que ser. el encargado de la empresa 1:N

Tema 3 Modelo Relacional

1.- El Modelo Relacional
    1.1.- Las Relaciones en el modelo Relacional
    1.2.- Conceptos necesarios para pasar del Modelo Conceptual (E/R) al Lógico (Relacional)
2.- Transformación de un diagrama E/R al modelo Relacional
3.- Normalización
4.- MySQL Workbench

Del Modelo E/R al Modelo Relacional 4 Fases Transformación

La fase de diseño lógico es aquella en la que se transforma el esquema conceptual (reflejado en un diagrama Entidad/Relación) en el modelo de datos específico del SGBD seleccionado (en nuestro caso, el modelo relacional).

Este paso se realiza en dos fases:
–Transformación del MER a un modelo independiente del SGBD concreto.
–Elaboración de las sentencias del lenguaje de definición de datos (DDL) del SGBD, que se completarán con las decisiones tomadas en el diseño físico posterior.

Una vez obtenido el esquema relacional independiente del SGBD, conviene realizar un refinamiento del mismo, centrándose en los siguientes aspectos:

–Establecer las claves primarias.
–Añadir las restricciones de integridad.
–Establecer la posibilidad de valores nulos.
–Normalizar el esquema obtenido.
–Definir las reglas de inserción, actualización y borrado.
–Revisar y validar el esquema con los usuarios.
–Documentar la solución final

El Modelo E/R Formulado por P.P. Chen en 1976. Es un Modelo de datos que representa un esquema de base de datos mediante entidades y asociaciones. Describe una base de datos de una forma sencilla y global. Se realiza a partir de los requisitos de datos que debe cumplir una base de datos.

El Modelo Relacional Formulado por Edgar F. Codd En 1970 propone un nuevo modelo de datos basado en la teoría de conjuntos y en el concepto matemático de relación. La estructura lógica principal son tablas o relaciones. Cada relación tiene un número fijo de columnas o atributos (esquema o intensión) y un número variable de filas o tuplas (extensión)
Una BD relacional está compuesta por varias tablas o relaciones.

1.- El modelo Relacional

La BD es como un conjunto de Relaciones que se pueden operar mediante el algebra relacional. El modelo Relacional es independiente de la forma en que se almacenan los datos y de la forma de representarlos, de forma que la BD se puede implementar en cualquier SGBD y los datos se pueden gestionar utilizando cualquier aplicación gráfica.

1.1.- Las Relaciones en el modelo Relacional

Una Relación es un conjunto de atributos, cada uno perteneciente a un dominio y con un Nombre que identifica la relación.
El conjunto de tuplas (filas) representa el **cuerpo de la relación** y el conjunto de atributos y el nombre representa el **Esquema**.
El **Grado** de la tabla es el **número de campos** que posee.
Cardinalidad es el número de tuplas concretas que almacena.
Valor. Viene representado por la intersección entre una fila y una columna. **Valor Null**. Representa la ausencia de información.

Conceptos

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:

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

1.2.- Conceptos necesarios para pasar del Modelo Conceptual (E/R) al Lógico (Relacional)

Atributo: Características de una entidad.

Dominio: Conjunto de valores permitidos para un atributo. Es el conjunto finito de valores homogéneos (todos del mismo tipo) y atómicos (son indivisibles), que puede tomar cada atributo. Todos los dominios tienen un nombre y un tipo de datos asociado.
Existen dos tipos de dominios:
– Dominios generales. Son aquellos cuyos valores están comprendidos entre un máximo y un mínimo. Por ejemplo, el Código postal, que está formado por todos los números enteros positivos de 5 cifras.
– Dominios restringidos. Son los que pertenecen a un conjunto de valores especifico. Por ejemplo, Sexo, que puede tomar los valores H o M.

•Nº de empleado (NUMEMP), Apellido (APELLIDO), Nº de departamento (NUMDEP), Salario (SALARIO).
•**La clave candidata** es el Nº de empleado. El apellido se puede repetir, así que no se le considera candidata. Como solo hay una se escoge primaria el Nº de empleado.
•El atributo Nº de departamento es **clave ajena**; relaciona las tablas TEMPLE y TDEPART.

Esquema de la base de datos

Una base de datos relacional es un conjunto de relaciones normalizadas. Para representar un esquema de una base de datos se debe dar el nombre de sus relaciones, los atributos de estas, los dominios sobre los que se definen estos atributos, las claves primarias y las claves ajenas.

En el esquema los nombres de las relaciones aparecen seguidos de los nombres de los atributos encerrados entre paréntesis. Las claves primarias son los atributos subrayados, y las claves ajenas se representan mediante diagramas referenciales.

Al realizar un Modelo de datos de un Sistema, se han de reflejar determinadas condiciones o políticas de negocio que se deben cumplir. Hay dos grupos:
- Restricciones Inherentes: Se derivan de la propia definición del modelo, no dependen del diseñador.
- Restricciones de Usuario o Semánticas: Son impuestas por el diseñador en función de los requisitos del sistema a modelar.

Restricciones Inherentes

No pueden haber dos tuplas iguales. No pueden haber dos ocurrencias o filas de una relación con el mismo valor en todos los campos.

El orden de las tuplas no es significativo.

El orden de los atributos (columnas) no es significativo.

Cada atributo solo puede tomar un único valor del dominio. No se admiten los grupos repetitivos, e.d., dos valores para un mismo campo.

Se debe cumplir la regla de la integridad de la entidad:"Ningún atributo que forme parte de la clave primaria puede tomar un valor desconocido o nulo".

Clave: Conjunto de atributos que identifican de forma única una ocurrencia de entidad, pueden ser simples o compuestas. Tipos:

- Clave Candidata: Conjunto mínimo de atributos que identifican unívoca y mínimamente cada tupla en una relación. Pueden haber varias.
- Clave Primaria: Es la clave Candidata elegida.
- Clave Alternativa: Claves candidatas que no han sido elegidas como primaria.
- Clave Foránea o Ajena: Conjunto de atributos de una relación cuyos valores han de coincidir con los valores de la clave primaria de otra relación (ambos deben estar definidos sobre los mismos dominios).

Restricciones Semánticas

Las Restricciones de Usuario o de semántica son las Condiciones que deben cumplir los datos:

- Restricciones de **clave** (identifican de forma única una entidad)
- Restricciones de **valor único** (UNIQUE)
- Restricciones de **integridad referencial**(*Nota1). Se da cuando una tabla tiene una referencia a algún valor de otra tabla. En este caso la restricción exige que exista el valor referenciado en la otra tabla.
- **Restricciones de dominio**. Exige que el valor que pude tomar un campo esté dentro del dominio definido.
- **Restricciones de verificación** (CHECK). Permite comprobar si el valor de un atributo es válido conforme a una expresión. Cada vez que se realice una inserción o una actualización de datos se comprueba si los valores cumplen la condición. Rechaza la operación si no se cumple.
- Restricciones de **valor nulo**. Si un atributo se admite como NULL es que es opcional.
- **Disparadores o triggers**. Son procedimientos que se ejecutan para hacer una tarea concreta en el momento de insertar, borrar o modificar información en una tabla.
- Restricciones genéricas adicionales o **aserciones** (ASSERT). Permite validar cualquiera de los atributos de una o varias tablas, en este caso en lugar de afectar a una relación como CHECK, puede afectar a dos o mas relaciones. La condición se establece sobre elementos de distintas relaciones. Pueden implicar a subconsultas en la condición. La definición de una aserción debe tener un nombre. Tiene vida por si misma.

Integridad Referencial

I**ntegridad referencial o restricción de clave ajena (FOREIGN KEY**). Se utiliza para enlazar relaciones, mediante claves ajenas, de una base de datos. La integridad referencial indica que los valores de la clave ajena en la relación hijo se corresponden con los de la clave primaria en la relación padre.
Además de definir las claves ajenas **hay que tener en cuenta las operaciones de borrado y actualización** que se realizan sobre las tuplas de la relación referenciada.

Las posibilidades son las siguientes:

•**Borrado y/o modificacion en cascada (CASCADE).** El borrado o modificación de una tupla en la relación padre (relación con la clave primaria) ocasiona un borrado o modificación de las tuplas relacionadas en la relación hija (relación que contiene la clave ajena). En el caso de empleados y departamentos, si se borra un departamento de la tabla TDEPART se borraran los empleados que pertenecen a ese departamento.

• **Borrado y/o modificación restringido (RESTRICT)**. En este caso no es posible realizar el borrado o la modificación de las tuplas de la relación padre si existen tuplas relacionadas en la relación hija. Es decir, no podría borrar un departamento que tiene empleados.

•**Borrado y/o modificación con puesta a nulos (SET NULL).** Esta restricción permite poner la clave ajena en la tabla referenciada a NULL si se produce el borrado o modificación en la tabla primaria o padre. Así pues, si se borra un departamento, a los empleados de ese departamento se les asignara NULL en el atributo NUMDEPT.

•**Borrado y/o modificación con puesta a valor por defecto (SET DEFAULT)**. En este caso, el valor que se pone en las claves ajenas de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla

2.- Transformación de un diagrama E/R al modelo Relacional

•El grafo relacional es la representación de un sistema mediante un conjunto de relaciones vinculadas entre sí por una o varias claves ajenas, a partir del cual podremos crear la B.D. en nuestro Sistema Informático con la ayuda del SGBD elegido. Lo que se pretende es pasar de describir conceptualmente el mundo mediante entidades y relaciones, a describirlo lógicamente mediante tablas.

•Las reglas básicas para transformar un esquema conceptual E-R a un esquema relacional son las siguientes:

– Toda entidad se transforma en una tabla.

– Todo atributo se transforma en columna dentro de una tabla.
– El identificador único de la entidad se convierte en clave primaria.
– Toda relación N:M se transforma en una tabla que tendrá como clave primaria la concatenación de los atributos clave de las entidades que asocia.

•Claves Principales: Los atributo/s que forman la clave principal se subrayan.
•Claves Secundarias: doble subrayado.
•Atributos opcionales: *
•Claves Ajenas: trazo discontinuo
•Opciones de Borrado(B/D) y Modificación(M/U): se indican con las letras:
M:C /UC=> Modificación en cascada
M:N /UN=> Modificación con puesta a NULL
M:D /UD=> Modificación con valor por defecto
M:R /UR=> Modificación Restringida

Reglas de Transformación. Resumen

El paso del modelo conceptual, basado en el modelo Entidad/Relación, al modelo relacional se basa en una serie de reglas que, seguidas de forma precisa y sistemática, garantizan la finalización del proceso de forma completa y fiable sin pérdida semántica.

Los pasos a seguir para la obtención del modelo lógico a partir del modelo conceptual son los siguientes:

–Transformación de atributos.
–Transformación de entidades.
–Transformación de relaciones.
–Transformación de jerarquías y agregaciones.
–Normalización.

Utilización de atributos atómicos.

El modelo relacional impone una condición sobre las características que debe poseer cada columna de una tabla relacional. Todas las columnas poseen datos simples, también llamados atómicos, de manera que cada uno de ellos representa una información unitaria y que no puede descomponerse en bloques de información más pequeños, (al menos desde el punto de vista del S.G.B.D.). Por ejemplo el Teléfono que es un atributo múltiple, pues consiste en un mismo atributo atómico Teléfono, duplicado un número indeterminado de veces, no tiene representación correspondiente en el modelo relacional, por lo que es necesario algún tipo de transformación que convierta atributos múltiples en atributos atómicos.
Un atributo **compuesto es aquél que puede dividirse en bloques de información más pequeños,** pero a diferencia de los múltiples, estos bloques pequeños no son homogéneos, sino diferentes entre sí. Por ejemplo el Domicilio de algunas entidades. Normalmente no es necesario distinguir entre calle, número, portal, planta, etc. Si en nuestro diseño deseamos distinguir entre

todos esos componentes del Domicilio entonces la única solución es crear atributos atómicos para cada uno de ellos.

Esta transformación consiste en crear una entidad débil llamada Teléfonos, y relacionarla con Clientes a través de una relación débil. Esta nueva entidad poseerá un único atributo atómico Número de Teléfono. El atributo múltiple se convierte en una entidad, ahora ya con atributos atómicos, de manera que cada Cliente se relacionará con 0, 1 ó muchas instancias de esta nueva entidad.
CLIENTES(NIF, nombre, apellidos, dir , ...)
TELEFONOS (NIFCliente, NumTelefono)
FK DC/UC

Conversión de entidades y relaciones a tablas.

Una vez preparados los atributos de las entidades y relaciones, la conversión del diagrama E-R al modelo relacional pasa por dos etapas: una en la que se convierten las entidades, y otra en la que se convierten las relaciones. Pero, las tablas que se van obteniendo no adoptan su forma definitiva hasta que se ha acabado el proceso.

Existe una correspondencia directa entre el concepto de entidad del diagrama E-R (una vez eliminados los atributos múltiples y los compuestos), y el concepto de tabla relacional.

2.1- Transformación de Entidades, Atributos y Dominios

En el modelo relacional, los atributos de las entidades y relaciones del MER se transforman en atributos del modelo relacional, teniendo en cuenta los casos:

Atributos compuestos: Se descomponen en atributos simples. Ejemplo: atributo Dirección.

Atributos multivaluados: Dan lugar a una nueva relación cuya clave primaria es la concatenación de la clave primaria de la entidad en la que se sitúa el atributo multivaluado más el nombre del atributo multivaluado. En ocasiones, si el atributo multivaluado no permite repeticiones, es suficiente éste como clave primaria.

Atributos Obligatorios: Restricción NOT NULL.

Atributos Opcionales: Atributos que pueden tomar valor NULL

Atributos Identificador Principal: Atributo/s que forman la clave primaria **PRIMARY KEY.**

Atributos Identificador Alternativo: Atributo/s con la restricción UNIQUE.

Atributos Derivados: Atributo/s que se obtienen como resultado de algún cálculo sobro otros atributos.
Dominios: Se transforman en los dominios del modelo Relacional.

2.2- Transformación de Interrelaciones

Conversión de relaciones binarias a tablas.

Conversión de relaciones es-un y relaciones débiles.

•La conversión de relaciones es-un y relaciones binarias débiles en general, no conlleva realmente la aplicación de ninguna regla, ya que la propia conversión de la entidad débil en tabla convierte automática e implícitamente la relación débil también.
•Supongamos por ejemplo, el diagrama que nos permite representar las facturas propias de cualquier negocio. Dado que el número de líneas de detalle de una factura es indeterminado, es necesario crear una relación débil que relacione cada factura con los detalles que en ella se facturan:

Al convertir las entidades Facturas y Líneas de Detalle en sus tablas correspondientes, se observa que la tabla de Líneas de Detalle hereda los atributos que forman la clave de Facturas. Se deberá tener la opción de borrado en cascada, e.d., no podrá existir ninguna ocurrencia de la entidad hija que no esté relacionada con una entidad padre.

2.2.1.- Transformación de Interrelaciones N:M = NUEVA TABLA

Estas interrelaciones dan lugar a una relación cuya clave será la concatenación de los Identificadores Principales de ambas entidades. Por tanto, los atributos que forman la clave primaria de la relación serán claves ajenas respecto a las relaciones en las que estos atributos son claves primarias. Si la interrelación tiene atributos, éstos pasarán a la nueva tabla generada.

En la práctica, es necesario especificar que ocurre en los casos de borrado y modificación de la clave primaria de referencia, indicando la acción correspondiente:
– Restricción.
– Asignación de valor nulo o valor por defecto.
– Operación en cascada.
– Ejecución de un procedimiento de usuario.

2.2.2.- Transformación de Interrelaciones 1:N

DOS OPCIONES:

Si hay un (0,1) = NUEVA TABLA
Si no hay = PROPAGACIÓN HACIA N

Cuando la relación que se desea convertir es del tipo 1:N hay dos posibles soluciones:
•**1/ Propagar la clave de la entidad** que actúa con participación máxima 1 a la tabla correspondiente a la entidad que actúa con participación máxima N, así como los atributos propios de la relación. Esto se aplica cuando la cardinalidad es obligatoria, es decir, cuando tenemos cardinalidad (1,1) y (1,N)

Inconveniente: la propagación de la clave de una entidad a otra genera la aparición de una clave externa en la entidad de destino. Esto obliga a especificar el mecanismo de eliminación y modificación en función de la semántica del problema.

2/ Transformarla en una nueva relación que tendrá como atributos las claves de ambas entidades y los atributos propios de la nueva relación. Su clave será la clave de la entidad que interviene en la relación con N ocurrencias.
Es recomendable cuando: La relación tiene atributos propios. La cardinalidad es opcional, e.d., la cardinalidad mínima es 0 en, al menos, uno de los extremos.

2.2.3.- Transformación de Interrelaciones 1:1

OPCIONES:

(0,1) (0,1) = NUEVA TABLA
(1,1) (1,1) = PROPAGACIÓN EN CUALQUIER SENTIDO
(0,1) (1,1) = PROPAGACIÓN DE 1 A 0

Existen dos soluciones:
• **Transformar la relación en una tabla.** Si las entidades poseen cardinalidades (0,1), la relación se convierte en una tabla.
• **Propagar la clave.** Si una de las entidades posee cardinalidad (0,1) y la otra (1,1), conviene propagar la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad de cardinalidad (0,1). Si ambas entidades poseen cardinalidades (1,1), se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra. En este caso, también se pueden añadir los atributos de una entidad a otra, resultando una única tabla con todos los atributos de las entidades y de la relación, si los hubiera, eligiendo como clave primaria una de las dos.
Si la relación tiene atributos, se propagan con la clave.

Relaciones reflexivas o recursivas

Normalmente = DOS TABLAS (Entidad y relación)

Son relaciones en las que participa un tipo de entidad. Para convertir una relación reflexiva a tabla hay que tener en cuenta sobre todo la cardinalidad. Lo normal es que toda relación reflexiva se convierta en dos tablas, una para la entidad y otra para la relación. Se pueden presentar los siguientes casos:

- Si la relación es **1:1, la clave de la entidad se repite**, con lo que **la tabla resultante tendrá dos veces ese atributo,** una como clave primaria y otra como clave ajena de ella misma. **No se crea la segunda tabla.**

- Si la relación es **1:M**, podemos tener dos casos:
  a) Caso de que la entidad muchos sea siempre obligatoria se procede como en el caso 1:1.

b) Si no es obligatoria, se crea una nueva tabla cuya clave será la de la entidad del lado muchos, y además se propaga la clave a la nueva tabla como clave ajena.

Si es **N:M,** se trata **igual que en las relaciones binarias**. La tabla resultante de la relación contendrá dos veces la clave primaria de la entidad del lado muchos, mas los atributos de la relación si los hubiera. La clave de esta nueva tabla será la combinación de las dos.

Conversión de relaciones no binarias a tablas

Las relaciones ternarias, cuaternarias y n-arias que unen más de dos relaciones se transforman en una tabla que contiene los atributos de la relación más los identificadores de las entidades relacionadas. La clave la forman todas las claves externas.

Generalizaciones:

•El modelo relacional no dispone de mecanismos para la representación de las relaciones jerárquicas, por lo tanto, se tienen que eliminar. Para ello hay que tener en cuenta:

– La especialización que los subtipos tienen respecto a los supertipos, es decir, los atributos diferentes que tengan asociados cada uno de los subtipos, que son los que se diferencian con el resto de atributos de los otros subtipos.
– El tipo de especialización que representa el tipo de relación jerárquica: total o parcial exclusiva, y total o parcial solapada.
– Otros tipos de relación que mantengan tanto los subtipos como el supertipo.
– La forma en la que se va a acceder a la información que representan tanto el supertipo como el subtipo.

Para pasar estas relaciones al modelo relacional se aplicará una de las siguientes reglas:

Eliminación de los subtipos: Los atributos y relaciones de cada subtipo se transfieren al supertipo. Integrar todas las entidades en una única eliminando a los subtipos. Esta nueva entidad contendrá todos los atributos del supertipo, todos los de los subtipos, y los atributos discriminativos para distinguir a qué subentidad pertenece cada atributo.

Todas las relaciones se mantienen con la nueva entidad. Esta regla puede aplicarse a cualquier tipo de jerarquía. La gran ventaja es la simplicidad pues todo se reduce a una entidad. El gran inconveniente es que se generan demasiados valores nulos en los atributos opcionales propios de cada entidad.

Inconvenientes: Proliferación de valores nulos en los atributos procedentes de los subtipos.
Cuando se pretende consultar un subtipo concreto, se accede a información no requerida (de los restantes subtipos). Se recomienda cuando los subtipos tienen pocos atributos

Eliminación del supertipo, transfiriendo los atributos del supertipo a cada uno de los subtipos. Las relaciones del supertipo se consideran para cada uno de los subtipos. La clave genérica del supertipo pasa a cada uno de los subtipos. Las relaciones en las que aparece el supertipo se

aplican a cada uno de los subtipos. Solo puede ser aplicada para jerarquías totales y exclusivas. Inconvenientes que presenta esta transformación:

– Se crea redundancia en la información, pues los atributos del supertipo se repiten en cada uno de los subtipos.
– El numero de relaciones aumenta, pues si el supertipo tiene relaciones, estas pasan a cada uno de los subtipos.

Inconvenientes:
–Se introduce redundancia.
–Se incrementa el numero de relaciones.
–Se pierde semántica, de forma que hay que acceder a varias tablas para recuperar la información común.
Se recomienda cuando el supertipo tiene pocos atributos y participa en pocas relaciones.

Supertipo y Subtipos: Insertar una relación 1:1 entre el supertipo y cada uno de los subtipos. Los atributos se mantienen y cada subtipo se identificara con la clave ajena del supertipo. El supertipo mantendrá una relación 1:1 con cada subtipo. Los subtipos mantendrán, si la relación es exclusiva, la cardinalidad mínima 0, y si es solapada 0 ó 1.

Utilizar una relación para representar al supertipo y tantas relaciones como subtipos haya (con la clave de la entidad padre). Habrá que añadir un atributo que indique el tipo de entidad al que se hace referencia. Se utiliza cuando:
•los subtipos tienen atributos dispares y/o interrelaciones diferentes
•Incorporar mayor semántica en el grafo relacional

Será necesario implementar las restricciones semánticas necesarias a través de CHECKS o DISPARADORES.