



UF1469: SGBD e instalación

Certificado de Profesionalidad

IFCT0310 - Administración de bases de datos



IFCT0310 > MF0224_3 > UF1469

SGBD e instalación.
IFCT0310

Rafael Ángel Prieto de Lope

ic editorial

SGBD e instalación. IFCT0310
Autor: Rafael Ángel Prieto de Lope

1ª Edición

© IC Editorial, 2014

Editado por: IC Editorial

C.I.F.: B-92.041.839

c/ Cueva de Viera, 2, Local 3 Centro Negocios CADI

29200 ANTEQUERA, Málaga

Teléfono: 952 70 60 04

Fax: 952 84 55 03

Correo electrónico: iceditorial@iceditorial.com

Internet: www.iceditorial.com

IC Editorial ha puesto el máximo empeño en ofrecer una información completa y precisa. Sin embargo, no asume ninguna responsabilidad derivada de su uso, ni tampoco la violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Mediante esta publicación se pretende proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para **IC Editorial** ninguna forma de asistencia legal, administrativa ni de ningún otro tipo.

Reservados todos los derechos de publicación en cualquier idioma.

Según el Código Penal vigente ninguna parte de este o cualquier otro libro puede ser reproducida, grabada en alguno de los sistemas de almacenamiento existentes o transmitida por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro, sin autorización previa y por escrito de IC EDITORIAL; su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes

intencionadamente reprodujeran o plagiaran, en todo o en parte, una obra literaria, artística o científica.

ISBN: 978-84-16433-36-0

Nota de la editorial: IC Editorial pertenece a Innovación y Cualificación S. L.

Presentación del manual

El **Certificado de Profesionalidad** es el instrumento de acreditación, en el ámbito de la Administración laboral, de las cualificaciones profesionales del Catálogo Nacional de Cualificaciones Profesionales adquiridas a través de procesos formativos o del proceso de reconocimiento de la experiencia laboral y de vías no formales de formación.

El elemento mínimo acreditable es la **Unidad de Competencia**. La suma de las acreditaciones de las unidades de competencia conforma la acreditación de la competencia general.

Una **Unidad de Competencia** se define como una agrupación de tareas productivas específica que realiza el profesional. Las diferentes unidades de competencia de un certificado de profesionalidad conforman la **Competencia General**, definiendo el conjunto de conocimientos y capacidades que permiten el ejercicio de una actividad profesional determinada.

Cada **Unidad de Competencia** lleva asociado un **Módulo Formativo**, donde se describe la formación necesaria para adquirir esa **Unidad de Competencia**, pudiendo dividirse en **Unidades Formativas**.

El presente manual desarrolla la Unidad Formativa, **UF1469: SGBD e instalación,**

perteneciente al Módulo Formativo, **MF0224_3: Administración de sistemas gestores de bases de datos,**

asociado a la unidad de competencia **UC0224_3:**
Configurar y gestionar un sistema gestor de bases de datos,

del Certificado de Profesionalidad **Administración de bases de datos.**

Índice

Portada

Título

copyright

Presentación

Índice

Capítulo 1 Sistemas gestores de base de datos

1. Introducción
 2. Introducción a la historia y evolución de los SGBD
 3. Enumeración y descripción de las funciones de los SGBD
 4. Clasificación de los SGBD
 5. Definición de la arquitectura de un SGBD atendiendo al modelo de tres capas propuesto por el comité ANSI-SPARC
 6. Resumen
- Ejercicios de repaso y autoevaluación

Capítulo 2 Diccionario de datos

1. Introducción
 2. Concepto
 3. Análisis de su estructura
 4. Justificación de su importancia como elemento fundamental en la instalación y mantenimiento de las bases de datos
 5. Resumen
- Ejercicios de repaso y autoevaluación

Capítulo 3 Análisis de la estructura funcional del SGBD

1. Introducción
 2. Procesos del SGBD
 3. Gestor de ficheros
 4. Procesador y compilador del DML
 5. Compilador del DDL
 6. Gestión de la BD
 7. Gestión de las conexiones y red
 8. Resumen
- Ejercicios de repaso y autoevaluación

Capítulo 4 Instalación de un SGBD

1. Introducción
2. Determinación de un SGBD a instalar en función de unos requerimientos planteados en un supuesto
3. Interpretación de la documentación de licencia de uso del SGBD
4. Identificación de las fuentes de documentación técnica. Interpretación de la documentación necesaria para la instalación
5. Identificación y verificación de los requisitos del computador necesarios para la instalación, así como los del sistema operativo
6. Descripción de los parámetros de configuración necesarios para la puesta en marcha del SGBD tanto a nivel del propio SGBD como del entorno en el que se instala
7. Selección de componentes lógicos adicionales que pueden ser de utilidad dependiendo del supuesto de instalación
8. Determinación de la ubicación y distribución idónea del software, los datos e índices dentro del computador

9. Cuando el SGBD soporta varios sistemas operativos y arquitecturas de computadores: identificar las ventajas e inconvenientes de seleccionar uno u otro
10. Identificación de los posibles juegos de caracteres y elementos de internacionalización más comunes así como los posibles problemas relacionados con estos
11. Realización de un supuesto práctico de instalación de un SGBD (y documentación del proceso) en el que se pongan de manifiesto las relaciones entre la arquitectura física del computador y las partes lógicas del SGBD
12. Resumen
Ejercicios de repaso y autoevaluación

Capítulo 5 Descripción de los mecanismos de comunicación del SGBD

1. Introducción
2. Configuración del acceso remoto a la base de datos en al menos un SGBD del mercado
3. Descripción de la comunicación cliente/servidor con el SGBD
4. Identificación de las diferencias de los medios de acceso cliente/servidor: socket, memoria compartida y TCP/IP
5. Identificación de los principales elementos que proveen de interoperabilidad al SGBD: ODBC, JDBC, etc.
6. Resumen
Ejercicios de repaso y autoevaluación

Bibliografía

Capítulo 1

Sistemas gestores de base de datos

1. Introducción

En el desarrollo de este capítulo se estudiarán los Sistemas de Gestión de Base de Datos (en adelante SGBD). Se verá su historia, el motivo de su aparición y sus funciones, estructura y principales arquitecturas. Los SGBD de mayor peso que existen en el mercado son *MySQL*, *Oracle* y *SQL Server*, aunque hay otros SGBD como por ejemplo: *PostgreSQL*, *SQLite*, *Firebird* o *FoxPro*.

Es importante diferenciar entre el concepto de base de datos entendido como una colección de datos relacionados entre sí, y un Sistema de Gestión de Base de Datos (SGBD) que es el *software* que controla y gestiona el acceso a la base de datos, y cuyo papel es cada vez más importante en el correcto funcionamiento de las aplicaciones actuales.

2. Introducción a la historia y evolución de los SGBD

Hace unos años las empresas e instituciones usaban para almacenar la información un sistema de almacenamiento de archivos, sistema que permitía almacenar y estructurar toda

la información además de poder realizar operaciones con estos archivos. Sin embargo, este sistema está en desuso debido a una serie de inconvenientes respecto a los actuales SGBD.

Algunos de estos inconvenientes son:

- **Redundancia e inconsistencia de datos:** pueden repetirse datos en diferentes archivos con el coste que ello provoca, además, se puede dar el caso de que esa duplicidad conlleve una incoherencia o inconsistencia entre los archivos duplicados.
- **Concurrencia:** cuando varios usuarios acceden a un mismo dato para modificarlo o borrarlo puede dar lugar a fallos de estado, sobre todo si el sistema operativo no es *Unix*.
- **Atomicidad:** esta es la propiedad que indica si una operación se ha realizado o no, y es clave en caso de fallo del sistema para saber si esa operación se ha realizado o por el contrario no lo ha hecho, en otras palabras, ciertas operaciones no pueden quedar a medias. Esta propiedad es complicada hacerla cumplir en un sistema de archivos tradicional.
- **Seguridad:** es difícil garantizar y mantener en el tiempo que ciertos archivos solo sean accesibles por determinados usuarios.
- **Dificultad en el acceso:** ciertas consultas pueden llegar a ser muy complicadas de realizar en este tipo de almacenamiento de la información.



Sabía que...

Los SGBD tienen su origen en la misión espacial Apolo, que llevó al hombre a la luna a finales de los años 60, debido a la inmensa cantidad de datos que había que

manejar en este tipo de misiones. Fue la empresa NAA la encargada de desarrollar este software.



Misión Apolo, primer SGBD

A finales de los 60 IBM se unió a NAA para desarrollar IMS (*Information Management System*), creando IBM su primer SGBD que al igual que el SGBD de NAA estaba basado en un modelo jerárquico, modelo que se estudiará más adelante.

También en estas mismas fechas nació el proyecto IDS (*Integrated Date Store*) dirigido por Charles Bachman, una

de las personas más influyentes en los recién nacidos SGBD. IDS se basaba en un sistema de red frente al sistema jerárquico de NAA. Este proyecto de IDS fue una primera aproximación a CODASYL, que era un estándar creado por el gobierno de EE. UU. y el mundo empresarial en 1971. Fue el nacimiento del primer SGBD en red. Al poner en funcionamiento estos SGBD en red aparecía el primer registro de la base de datos, que a su vez tenía punteros hacia otros registros, y así sucesivamente se podía ir aproximando al dato buscado. Ambos sistemas, en red y jerárquico, se englobaron posteriormente en base de datos de navegación.

En la década de los 70 se produjo una revolución en los SGBD: nace el concepto de base de datos relacional y el lenguaje *SQL*. Edgar Codd, descontento con la eficiencia de CODASYL, sobre todo a la hora de buscar un dato, observaba que las listas encadenadas no eran la mejor forma de almacenar la información. Codd creó un nuevo sistema en el que existían tablas con una clave que las enlazaba con otras tablas, nace el modelo relacional.

Posteriormente, el artículo escrito por Codd sobre el modelo relacional llegó a Eugene Wong y Michael Stonebraker. Ambos inician el proyecto INGRES que crea un lenguaje de acceso a datos *QUEL* y que posteriormente derivaría en *SQL*. IBM lanza su nueva generación de SGBD en 1975 basándose en el nuevo concepto relacional llamado *System R*. Este proyecto, antecesor a *Database2* (DB2), no es perfecto, presenta algunos inconvenientes principalmente a la hora de modelar el diseño. Una primera solución a esto la da el investigador Chen en 1976 con su propuesta de lenguaje de modelado denominado Entidad-Relación. Este modelo es usado hoy en día y está ampliamente aceptado para el diseño de base de datos.

En la década de los 80 nacen los SGBD orientado a objetos, ofreciendo muchas de las ventajas que este modelado tiene como lenguaje. También a finales de esta década dos investigadores norteamericanos muestran en una conferencia las ventajas de tener una pequeña base de datos, copia de la original, para mejorar las prestaciones. Este fue el origen de la indexación que hoy día prácticamente todos los SGBD incorporan.

Por último destacar los SGBD *NoSql* y a *XML*. El primero usa un modelado no relacional, normalmente basado en Clave-Valor y base de datos orientada a documentos. Por otro lado, las bases de datos *XML* que surgen en 2010, también con modelado *NoSql*, usan el lenguaje *XML* como formato de almacenamiento.



Actividades

1. Busque en internet al menos cuatro SGBD de software libre y otros cuatro SGBD de tipo comercial.
-

3. Enumeración y descripción de las funciones de los SGBD

Codd en 1985 publica las doce reglas que debe satisfacer cualquier SGBD relacional. Aunque estas reglas van enfocadas a los SGBD relacionales, se pueden extraer de ellas las funciones que todo SGBD, sea o no relacional, debe cumplir:

- **Almacenamiento, extracción y actualización datos:** es esta la función más importante que todo

SGBD debe ofrecer a los usuarios sin que este deba conocer los detalles técnicos de la implementación, como por ejemplo la organización de los archivos.

- **Soporte transacciones:** las transacciones son un conjunto de acciones llevadas a cabo sobre una base de datos por un único usuario o aplicación, de tal manera que todas estas acciones deben realizarse, o por el contrario ninguna, pero bajo ningún concepto pueden quedar a medias.

Ejemplo: un ejemplo muy ilustrativo de transacción se observa en las operaciones bancarias, si se hace una transferencia de una cuenta bancaria a otra se producen dos acciones, decremento del saldo de la cuenta origen y aumento del saldo de la cuenta destino. Como consecuencia, o se hacen ambas acciones o no se hace ninguna.

- **Catálogo accesible al usuario:** los usuarios autorizados deben tener acceso a la estructura de la base de datos. El catálogo del sistema es un componente de gran peso dentro de los SGBD. Este pueden contener:
 - Acceso de los usuarios a la base de datos.
 - Los elementos y tipo de acciones (**insert - update - delete - select**) que un usuario tiene asignado.
 - Nombres, tipos, tamaños y descripción de los diferentes elementos.
 - Estadísticas sobre transacciones, uso, accesos, etc.
- **Garantizar la concurrencia:** los SGBD deben controlar y garantizar la concurrencia.

Recuerde: la concurrencia sucede cuando se accede a un mismo recurso de forma paralela. Un SGB debe controlar que no existan anomalías cuando esto sucede.

- **Independencia de los datos:** existen dos tipos de independencia de los datos: lógica y física:
 - Independencia física: se deberían poder cambiar, por ejemplo, diferentes dispositivos de almacenamiento sin tener que modificar nada de la estructura lógica de la base de datos, de manera que el usuario pueda notar un cambio en las prestaciones pero nada más.
 - Independencia lógica: cambios en el esquema conceptual, es decir, añadir o eliminar nuevas entidades, atributos o relaciones, no deberían afectar a los usuarios que no tienen acceso a los elementos modificados. Esta independencia es muy complicada alcanzarla.
- **Seguridad:** un SGBD debe proporcionar un mecanismo para recuperar la base de datos en caso de la pérdida provocada por cualquier anomalía o accidente. Por otro lado se debe garantizar que solo los usuarios autorizados accedan a la base de datos, así como no permitir el acceso de un usuario a un elemento o función a la que no esté estrictamente autorizado.
- **Procesamiento distribuido:** un SGBD debe permitir integrarse, al menos, con varios DCM (*Data Communication Manager*) o gestores de comunicaciones, que es el *software* encargado de establecer una comunicación vía mensajes entre la máquina donde se encuentra el SGBD y el equipo remoto o equipo de una red de área local.
- **Integridad:** este concepto hace referencia a la coherencia y corrección de los datos almacenados en una base de datos. Esta integridad se suele expresar mediante restricciones. Por ejemplo, no permitir que el salario de un empleado supere los 100.000 € mensuales.



Actividades

2. Piense en uno o varios escenarios reales donde se pueda dar o usar la integridad de los datos, concurrencia, integridad, seguridad y transacciones.
 3. Si usase el SGBD en su equipo de manera local, ¿qué funciones de las descritas en el punto anterior no serían fundamentales?
-

4. Clasificación de los SGBD

Los SGBD pueden clasificarse en función del modelo de datos, dependiendo del número de usuarios al que da servicio (monousuario o multiusuario), atendiendo a si es un sistema distribuido o centralizado, y por último, a si la gestión de los procesos es multihilo y multiproceso.

4.1. Modelo de datos

Un **modelo** es una representación de la realidad. El modelo de datos debe permitir a los diseñadores de base de datos y a los usuarios finales comunicar e interactuar de forma precisa y no ambigua con la base de datos. Los modelos de datos tienen tres componentes:

- Una **parte estructural**, que está compuesta por un conjunto de reglas que definen como debe construirse una base de datos.
- Una **parte operacional**, que define las operaciones que pueden realizarse sobre los datos.

- Una **parte de restricciones**, para dar mayor integridad a los datos.

En la literatura técnica se han propuesto varios modelos de datos que se podrían clasificar en dos grandes grupos: los basados en objetos y los basados en registros.

Modelo de datos basados en objetos

Los modelos de datos basados en objetos utilizan conceptos como entidades, relaciones y atributos. La entidad es un objeto singular, como por ejemplo, una persona, un concepto, una cosa, etc. La relación es una asociación entre entidades, y los atributos son una propiedad relevante de la entidad. Algunos de los modelos de datos basados en objetos más destacados son:

- Entidad-Relación.
- Orientado a objetos.
- Funcional.

De estos, quizás el más popular para el diseño de base de datos es el modelo Entidad-Relación. Para representar este modelo la notación más usada es UML (*Unified Modeling Language*).

Entidad-Relación

Es probablemente el modelo más usado para el diseño de base de datos. Este modelo, creado por Chen en 1976, emplea tres conceptos básicos: entidad, atributos y relaciones.

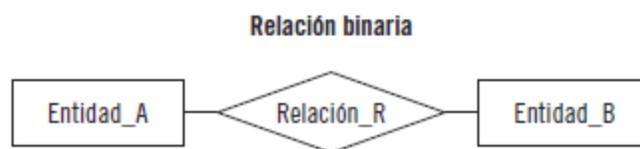
La entidad es cualquier objeto real o abstracto del cual merece la pena guardar información. Cada entidad cuenta con un conjunto de propiedades llamadas atributos, por último, lo que asocia a una

serie de entidades sería la relación, además, cada relación podría tener atributos.

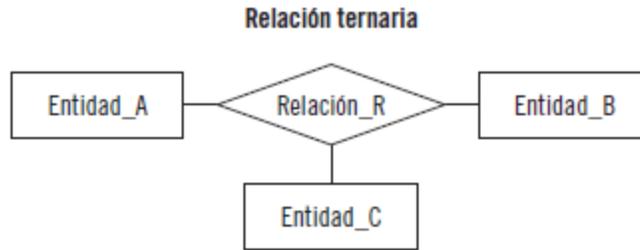
Cada entidad debe tener un atributo principal llamado clave principal o clave primaria. Este atributo lo selecciona el diseñador y debe servir para distinguir a cada una de las entidades de un conjunto. Un ejemplo típico de la clave principal podría ser la entidad "Persona", con varios atributos (nombre, apellidos, sexo, etc.) y donde la clave principal sería el NIF.

Las relaciones pueden ser n-arias, aunque en la práctica las formas más habituales son las relaciones binarias, ternarias o recursivas, siendo las binarias las más comunes y más usadas. En las relaciones binarias hay dos entidades y una relación entre ellas, en las ternarias, intervienen tres entidades y una relación, y por último, en las relaciones recursivas hay una única entidad y una relación. El aspecto sería:

■ Binaria:



■ Ternaria:

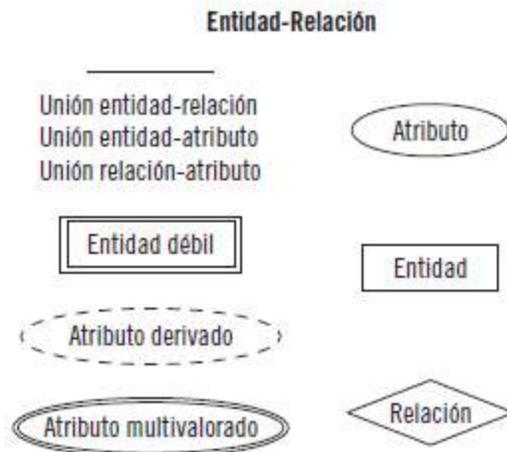


■ Recursiva:



Las relaciones introducen el concepto de cardinalidad, que expresa el número de entidades a las que otra entidad se puede asociar mediante un conjunto de relaciones. Pueden ser: uno-uno, uno-variantos, varios-uno y varios-variantos (1:1, 1:N, N:1 y N:N).

Los símbolos usados para representar de manera gráfica el modelo Entidad-Relación serían:



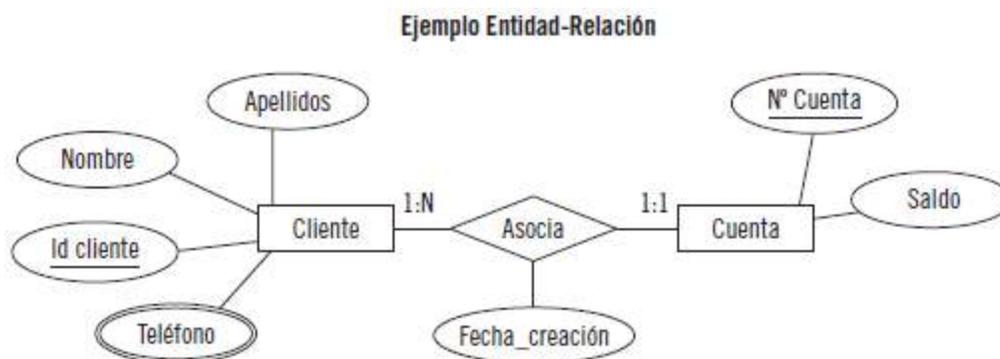
Algunos conceptos más asociados a este modelo son:

- **Entidad débil:** es aquella cuya existencia depende de otra. Por ejemplo, relacionado con el sector de la banca, un pago o cuota de un préstamo sería una entidad débil de préstamo, ya que si no existiese el préstamo, no tendría sentido el pago o cuota.
- **Atributo derivado:** es un atributo cuyo valor se genera a partir de otro atributo, por ejemplo, volviendo a la entidad “Persona”, el atributo “edad” sería derivado del atributo “fecha de nacimiento”.
- **Atributo multivalorado:** son aquellos atributos que, como su nombre indica, tienen más de un valor. Un ejemplo típico sería el número de teléfono.



Ejemplo

Un ejemplo sencillo de diagrama Entidad-Relación podría ser representar los clientes de una entidad bancaria y su cuenta o cuentas corrientes:



Se observan en el ejemplo las entidades “Cliente” y “Cuenta” con sus atributos, su clave primaria (“Id_cliente” y “Num_cuenta”) y atributo multivalorado

("Teléfono"). Además, la relación que los une, que este caso lleva un atributo ("Fecha_creación"), es binaria, y la cardinalidad se interpreta de la siguiente forma: un cliente tiene asociada una o varias cuentas (1:N), y una cuenta, en este ejemplo, tiene asociada a un solo cliente (1:1).



Actividades

4. Piense un ejemplo donde sería apropiado usar una entidad débil.
-

Orientado a objetos y funcional

Está basado en el concepto de orientación a objetos, y con este concepto nacieron los SGBDOO (Sistemas Gestores de Bases de Datos Orientados a Objetos). No se entrará en detalles sobre la orientación a objetos porque no es motivo de estudio de la presente materia.

En cuanto al modelo funcional, este es una técnica basada en objetos también, e incluye conceptos como la herencia, sobrecarga y acceso navegacional.

Modelo de datos basado en registros

Los modelos de datos basados en registros tienen una serie de registros de formato fijo. Cada tipo de registro define un número de campos, cada uno con una longitud también establecida. Los tres tipos principales de modelos de datos basados en registros son:

- Modelo de datos relacional.

- Modelo de datos en red.
- Modelo de datos jerárquico.

Modelo de datos relacional

Está basado en el concepto de las relaciones matemáticas. En el modelo relacional los datos y las relaciones se representan mediante tablas. Cada una de estas tablas tiene un número de columnas determinado con nombre único dentro de la propia tabla y cada columna representa los atributos o propiedades al igual que en el modelo de datos basado en objetos.



Ejemplo

Trata sobre los empleados de una entidad bancaria. En las columnas se indican varios aspectos relevantes sobre estos, como son el nombre, el apellido, la fecha de nacimiento y el DNI. En la tabla de más abajo se representan las entidades bancarias donde estos empleados podrían estar destinados.

Tabla de empleados				
Nombre	Apellido	Fecha_nac	Dni	Num_sucursal
Angie	García	05/07/1975	47483498E	12AB
Jesús	Fuentes	29/12/1988	40003498P	12AB
Javier	Morales	05/08/1974	42883407Y	458U
María Soledad	Rubio	01/02/1982	37483400H	89TY
Francisco Javier	Pérez	05/02/1979	44834698T	01MN

Tabla de sucursales

Num_sucursal	Calle	Ciudad
12AB	Los Palacios 1	Utrera
458U	Preciosa 2	Utrera
89TY	Rueda 5	Rota
01MN	Rubio 1	Torremolinos

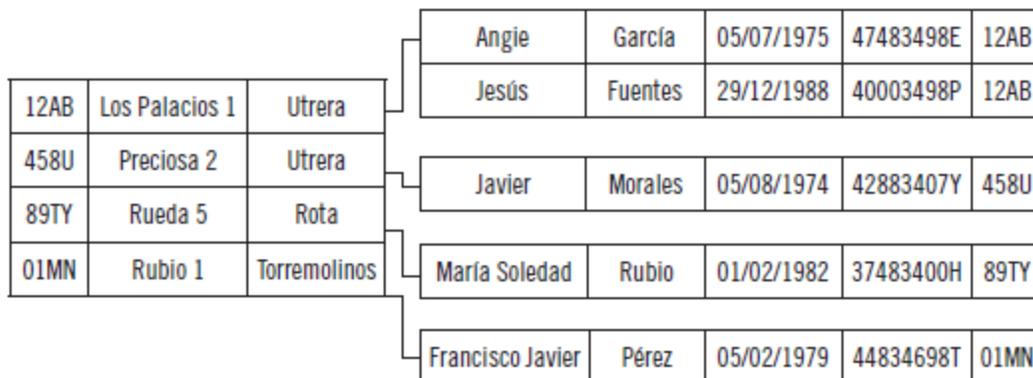
Ejemplo esquema relacional

Como se observa, para indicar una relación entre los empleados del banco y las sucursales bancarias se usa la columna "Num_sucursal".

Modelo de datos en red

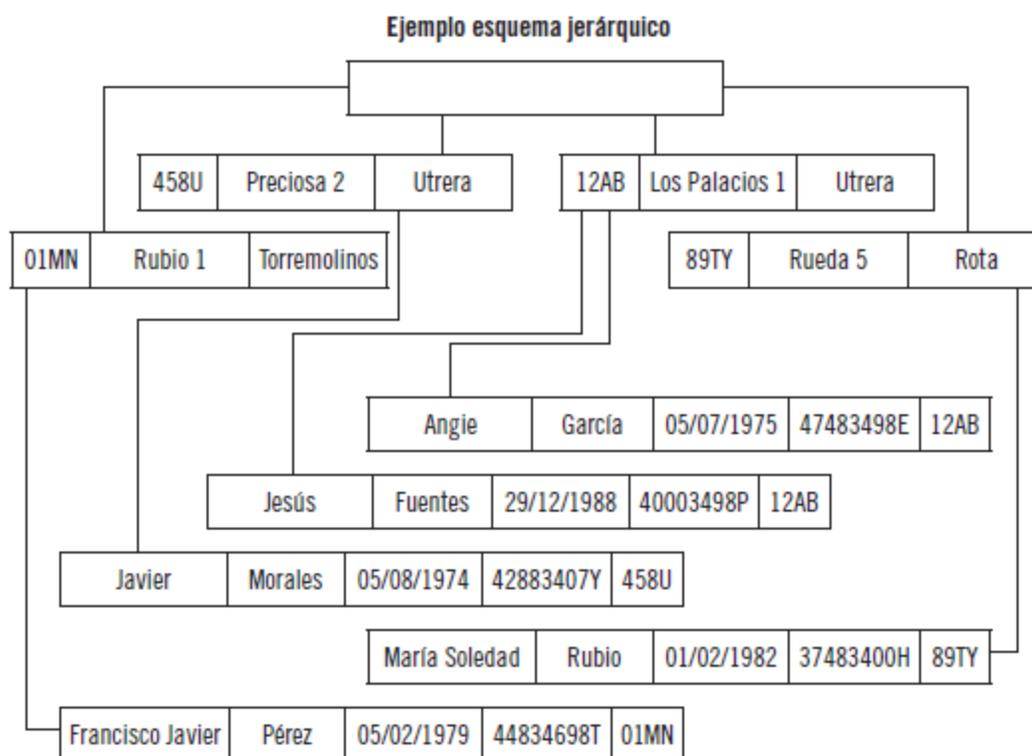
El modelo en red representa los datos como colecciones de registros, y las relaciones mediante conjuntos. Los registros se organizan como grafos. En la siguiente imagen se representa el modelo de datos en red:

Ejemplo esquema en red



Modelo de datos jerárquico

El modelo de datos jerárquico es un tipo restringido del modelo en red, por lo que los datos se representan como colecciones de registros y las relaciones como conjuntos, con la salvedad que este modelo solo permite que cada nodo tenga un solo padre.



4.2. Número de usuarios a los que da servicio: monousuario y multiusuario

En general los SGBD se comportan de manera que soportan más de un usuario, solo en el caso de ordenadores personales puede darse la situación de tener un SGBD monousuario, por lo que directamente se mostrarán los SGBD con arquitectura multiusuario. En concreto, se

analizarán las arquitecturas de cliente-servidor y servidor de archivos.

Arquitectura cliente-servidor

La arquitectura cliente-servidor es muy usada en los sistemas de información actuales: un cliente que solicita o necesita un servicio o recurso y un servidor que lo facilita. Generalmente, el servidor se encuentra situado en una máquina y el cliente o clientes en otra. En esta situación se puede dar el caso de que clientes y servidor estén en la misma red (LAN, red de área local), o por el contrario, que no estén en la misma red, lo que provocaría llevar a cabo por parte de la empresa/institución una política específica de seguridad.

Dentro de la arquitectura cliente-servidor se pueden destacar dos formas diferentes de implementarla, una con dos capas y otra con tres capas.

Arquitectura cliente-servidor dos capas

Es la arquitectura estándar del modelo cliente-servidor, donde el cliente presenta los datos al usuario y el servidor es el responsable de suministrarlos usando como canal una red.



Nota

La arquitectura cliente-servidor se puede implementar en dos, tres o más capas, aunque en este capítulo solo se profundizará en las arquitecturas de dos y tres capas, ya que aumentar el número de capas a cuatro o cinco no aportaría ninguna novedad respecto al SGBD.
