

Unidad 1:

Sistemas de almacenamiento de Información

Julia Triana

IES Juan José Calvo Miguel

Analizar los sistemas lógicos de almacenamiento y sus funciones.

Identificar los tipos de bases de datos según el modelo de datos utilizado.

Identificar los tipos de bases de datos según la ubicación de la información.

Valorar la utilidad de un sistema gestor de bases de datos.

Describir la función de cada elemento de un sistema gestor de bases de datos.

Clasificar los sistemas gestores de bases de datos.

Introducción

Ficheros

Bases de datos

SGBD. Funciones, componentes y tipos

BBDD centralizadas y BBDD distribuidas

- Necesidad de manejar gran cantidad de datos de una forma fácil, rápida y sencilla
- Imprescindible para la toma de decisiones
- Necesidad de gestionar los datos



- Evolución de los sistemas de almacenamiento

Fichero manual

Sistema de Ficheros

Sistema de BBDD

Fichero manual: conjunto de carpetas etiquetadas cuyo contenido estaba relacionado y se guardaban en un armario o archivador.

Sistema de Ficheros: Los datos necesarios se guardaban en ficheros en el ordenador y las aplicaciones accedían a ellos para obtener los informes que se solicitaban de cara a la toma de decisiones en la empresa.

Sistema de BBDD : Una base de datos es un conjunto de datos relacionados entre sí, organizados y estructurados, con información referente a algo. Cualquier cambio en la estructura de los datos no afectará a los programas de aplicación que los utilicen. La información es tratada utilizando los SGBD.

Introducción

Ficheros

Bases de datos

SGBD. Funciones, componentes y tipos

BBDD centralizadas y BBDD distribuidas

¿Qué son?

Conjunto de información relacionada con respecto a **un mismo aspecto** y que se guarda en dispositivos de almacenamiento (discos duros, lápiz usb, nube)

```
mysql> describe products;
```

Field	Type	Null	Key	Default	Extra
item_code	int(4)	NO			
item_name	varchar(50)	NO			
description	varchar(255)	NO			
category	varchar(50)	NO			
quantity	int(4)	NO			
price	float	YES		NULL	
image	blob	YES		NULL	
imagenname	varchar(50)	YES		NULL	
id	int(10)	NO		0	

```
9 rows in set (0.28 sec)
```




Code	Name	Continent	Region	Population	Capital
ABW	Aruba	North America	Caribbean	103000	129
AFG	Afghanistan	Asia	Southern and Central Asia	22720000	1
AGO	Angola	Africa	Central Africa	12878000	56
AIA	Anguilla	North America	Caribbean	8000	62
ALB	Albania	Europe	Southern Europe	3401200	34
AND	Andorra	Europe	Southern Europe	78000	55
ANT	Netherlands Antilles	North America	Caribbean	217000	33
ARE	United Arab Emirates	Asia	Middle East	2441000	65
ARG	Argentina	South America	South America	37032000	69
ARM	Armenia	Asia	Middle East	3520000	126

Datos , campo, registro, fichero

```
mysql> describe products;
```

Field	Type	Null	Key	Default	Extra
item_code	int(4)	NO			
item_name	varchar(50)	NO			
description	varchar(255)	NO			
category	varchar(50)	NO			
quantity	int(4)	NO			
price	float	YES		NULL	
image	blob	YES		NULL	
imagenname	varchar(50)	YES		NULL	
id	int(10)	NO		0	

```
9 rows in set (0.28 sec)
```



Code	Name	Continent	Region	Population	Capital
ABW	Aruba	North America	Caribbean	103000	129
AFG	Afghanistan	Asia	Southern and Central Asia	22720000	1
AGO	Angola	Africa	Central Africa	12878000	56
AIA	Anguilla	North America	Caribbean	8000	62
ALB	Albania	Europe	Southern Europe	3401200	34
AND	Andorra	Europe	Southern Europe	78000	55
ANT	Netherlands Antilles	North America	Caribbean	217000	33
ARE	United Arab Emirates	Asia	Middle East	2441000	65
ARG	Argentina	South America	South America	37032000	69
ARM	Armenia	Asia	Middle East	3520000	126

- **Datos:** información que necesitamos almacenar

Ej: Los datos personales de los clientes de una empresa, sin tratar.

- **Campo:** Es un carácter o conjunto de caracteres que tiene significado específico. Se utiliza para definir y guardar datos. Es la mínima unidad de información creada con sentido en sí misma.

Ej: Definiríamos un campo para guardar el nombre, en otro campo para el teléfono, etc.

- **Registro:** (Registro Lógico) Es un conjunto de campos lógicamente relacionados que describen una persona, lugar o cosa. Es también la unidad de tratamiento de los ficheros de datos.

Ej: Un registro lo formarían todos los datos/campos relativos a un cliente: su nombre, sus apellidos, su teléfono y su dirección.

- **Fichero:** Es un conjunto de registros relacionados.

Ej: El fichero estaría formado por los datos de todos los clientes del taller.

Registro lógico:

Es cada uno de los componentes del archivo. Tienen un conjunto de funciones que se acceden y se tratan de manera unitaria.

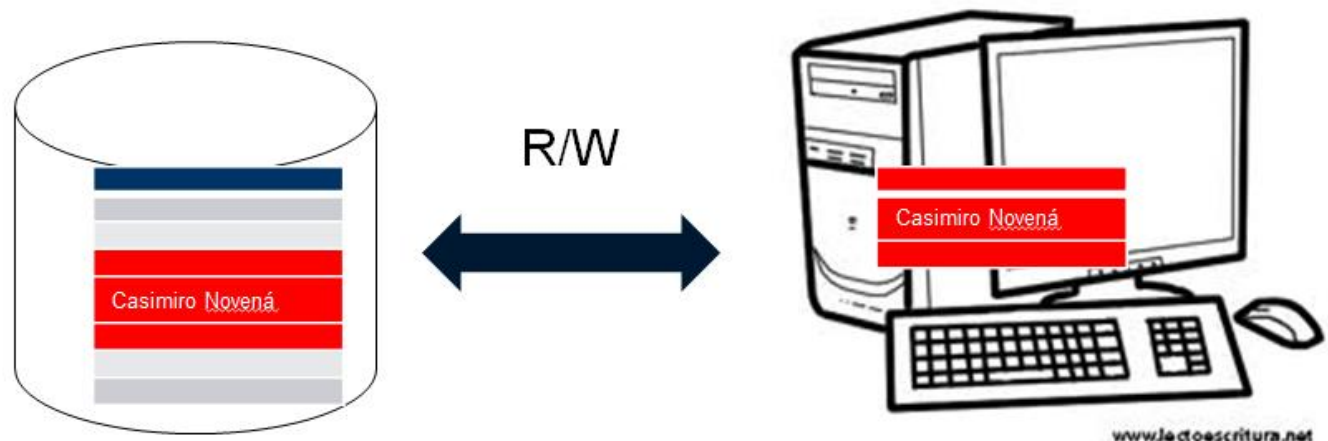
Componentes:

- Campos
- Subcampos
- Campo clave

Registro Físico o bloque:

Corresponde a la cantidad de información que se trasmite físicamente en cada operación.

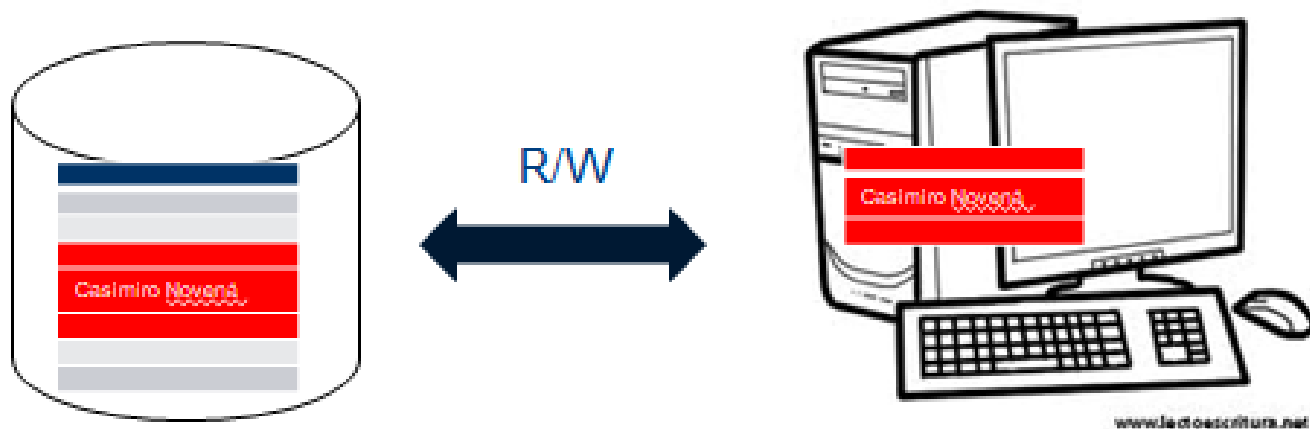
Es imposible tener todo el fichero en la memoria, por su tamaño, para procesarlo.



Factor de bloqueo o bloque de registros:

número de registros que entran en un bloque físico y que se transfieren en una operación de R/W.

Ej. Factor de bloqueo 3



Operaciones a realizar sobre ficheros

Abrir (open)

- Prepara el fichero para su proceso.

Cerrar (close)

- Cierra el fichero impidiendo su proceso inmediato.

Leer (read)

- Obtiene información del fichero.

Escribir (write)

- Graba información en el fichero.

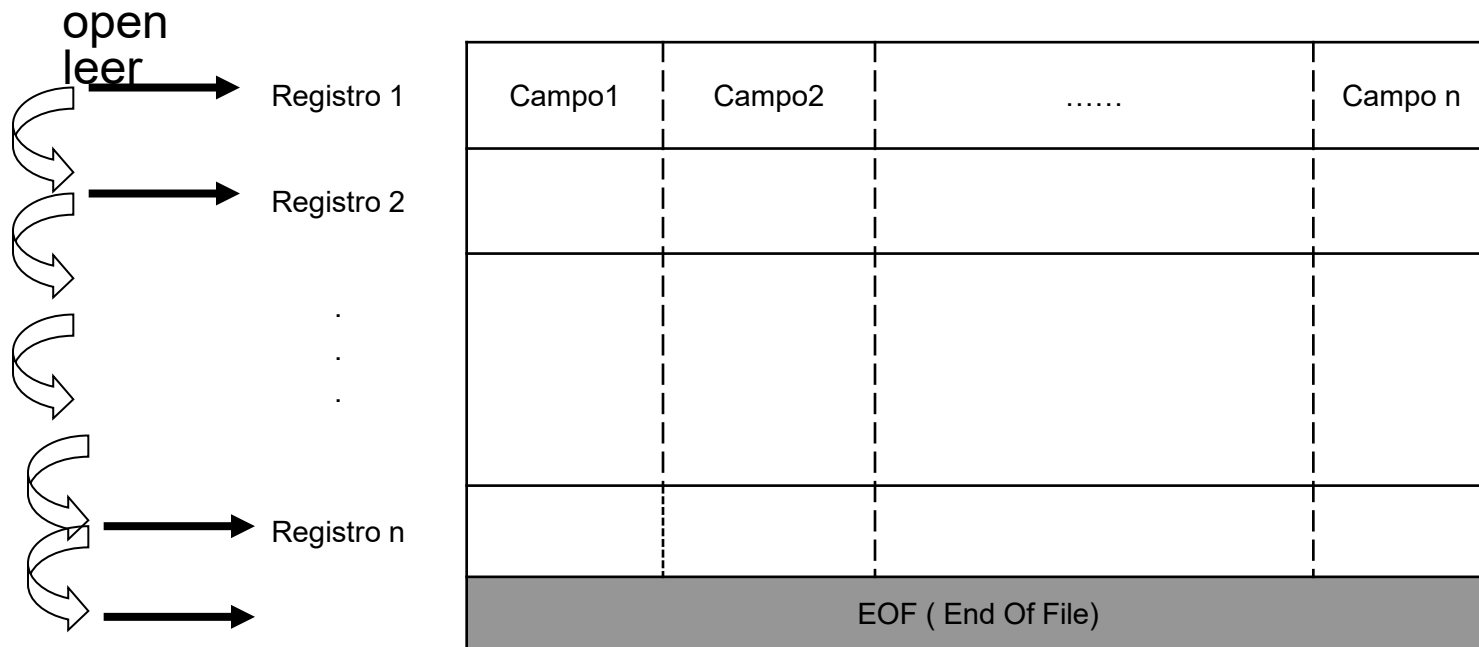
Posicionarse (seek)

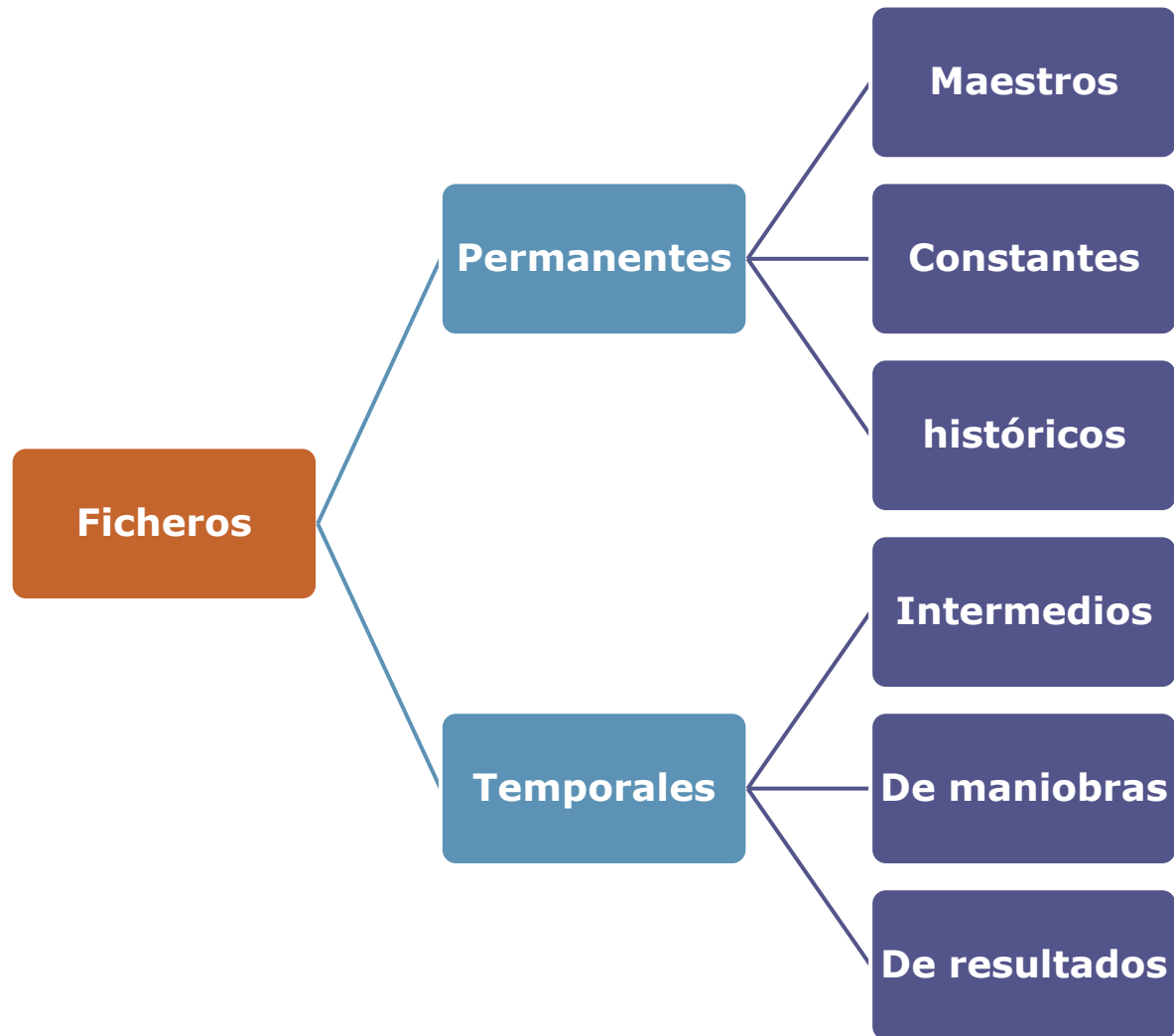
- Coloca el puntero de lectura en una posición concreta del mismo (no se puede realizar en todos los tipos de ficheros).

Fin de fichero (eof)

- Indica si hemos llegado al final del fichero.

Operaciones a realizar sobre ficheros





Ficheros permanentes: Se caracterizan por contener información relevante para una aplicación. Es decir, los **datos** necesarios para el funcionamiento de ésta. Tienen un periodo de permanencia en el sistema amplio. Estos se subdividen en:

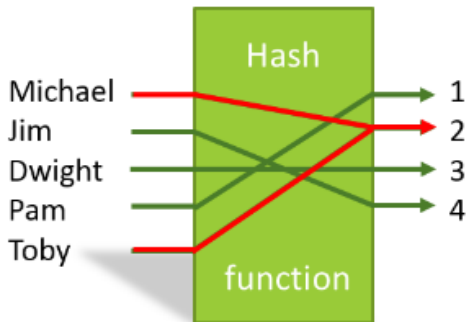
- **Ficheros maestros:** contienen el **estado actual de los datos** que pueden modificarse desde la aplicación. Es la parte central de la aplicación, su núcleo. Podría ser un archivo con los datos de los usuarios de una plataforma educativa.
- **Ficheros constantes:** son aquellos que incluyen **datos fijos** para la aplicación. No suelen ser modificados y se accede a ellos para realización de consultas. Podría ser un archivo con códigos postales.
- **Ficheros históricos:** contienen **datos que fueron considerados como actuales** en un periodo o situación anterior. Se utilizan para la reconstrucción de situaciones. Podría ser un archivo con los usuarios que han sido dados de baja en la plataforma educativa.

Ficheros temporales: Se utilizan para almacenar información útil para una parte de la aplicación, no para toda ella. Son generados a partir de datos de ficheros permanentes. Tienen un corto periodo de existencia. Estos se subdividen en:

- **Ficheros intermedios:** almacenan resultados de una aplicación que serán utilizados por otra.
- **Ficheros de maniobras:** almacenan datos de una aplicación que no pueden ser mantenidos en memoria principal por falta de espacio.
- **Ficheros de resultados:** almacenan datos que van a ser transferidos a un dispositivo de salida.

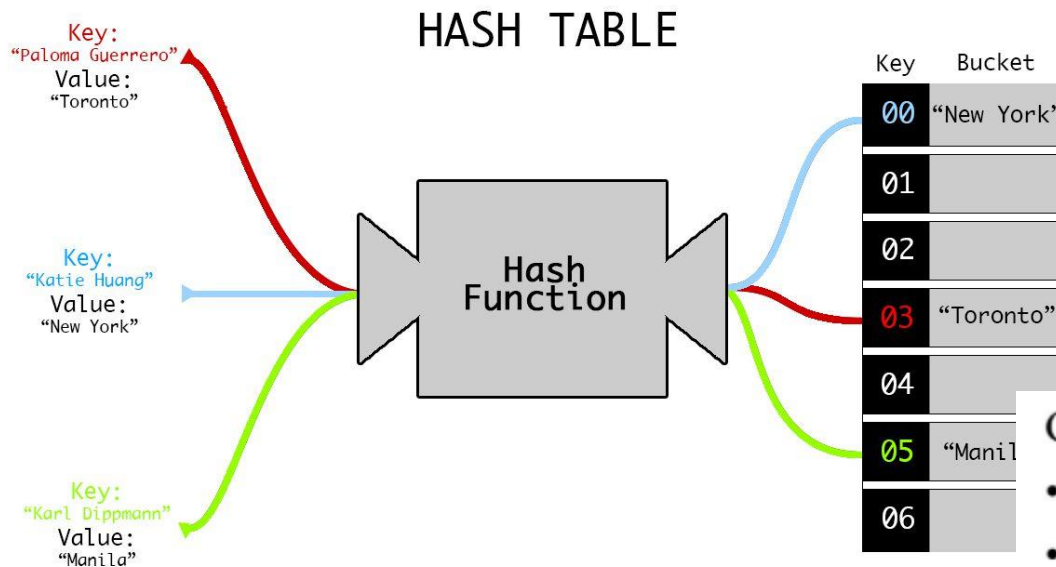
<http://bigdataanalyticsweb.blogspot.com>

Algoritmo de hashing



OBJECT → INTEGER

Algoritmo de hashing para posiciones de memoria



General idea:

- Get a large array
- Design a *hash function* h which converts keys into array indices .
- Insert (key, value) in the array at $h(\text{key})$ position.
- Can also be used in external memory (addresses instead of array indices).

- El algoritmo de transformación o **hashing** debe cumplir las siguientes condiciones:
 - Que sea **fácil de aplicar**, estableciendo una relación directa entre dirección lógica y dirección física.
 - Que deje el **mínimo número de huecos** posible, maximizando el espacio disponible en el dispositivo de almacenamiento.
 - Que las claves de registros diferentes nos den direcciones diferentes. Producir el menor número de registros que con distintas claves creen las mismas direcciones de almacenamiento. Cuando a partir de dos o más claves diferentes se obtiene la misma dirección se dice que se producen **sinónimos** y que esos registros producen **colisiones**. En este caso solo uno de ellos puede ser almacenado en esa dirección y habrá que prever algún procedimiento para calcular la posición en que se tiene que grabar el otro registro. **Por ejemplo una zona de overflow.**

Introducción

Ficheros

Bases de datos

SGBD. Funciones, componentes y tipos

BBDD centralizadas y BBDD distribuidas

Tipos de sistemas de información

Cada programa almacena y utiliza sus propios datos de forma independiente.

**Orientados
a los procesos**

Sistemas de información orientados a procesos



**Orientados
a los datos**

Sistemas de información orientados a datos



SI orientados a procesos – Inconvenientes (I)

Cada programa almacena y utiliza sus propios datos de forma independiente.

**Orientados
a los procesos**



- Dependencia de los datos a nivel físico.
- Datos redundantes.
- Datos inconsistentes.
- Dificultad para el acceso simultáneo a los datos.
- Difícil acceso a los datos.
- Coste de almacenamiento elevado.
- Tiempos de procesamiento elevados.
- Dificultad para administrar la seguridad del sistema.

SI orientados a procesos

Inconvenientes (II)

Sistemas de información orientados a procesos



Redundancia.

La misma información es almacenada varias veces en la misma base de datos

PERSONAL		
Cedula	Nombre	Dirección
7492837	Ana	Calle 1
9836384	Pedro	Avenida 3
8038464	José	Urb. Los ...

PROFESORES		
Cedula	Nombre	Dirección
7492837	Ana Maria	Calle 55
9836384	Pedro	Avenida 3

Inconsistencia.

Consiste en que no todas las copias redundantes contienen la misma información

Orientados
a los datos

Ventajas

- Independencia de los datos y los programas
- Menor redundancia
- Integridad de datos
- Mayor seguridad en los datos
- Datos mas documentados
- Acceso a los datos mas eficiente
- Menor espacio de almacenamiento
- Acceso simultaneo a los datos

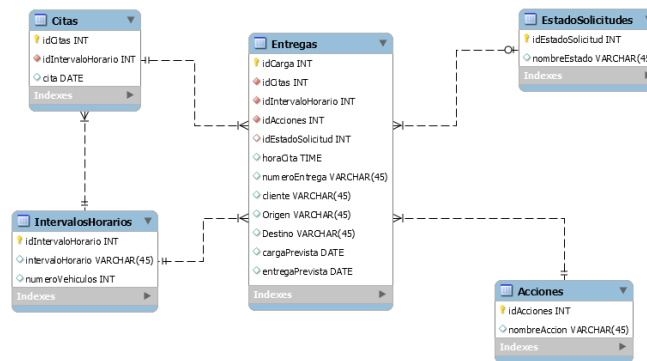


Desventajas

- Instalación costosa
- Requiere personal cualificado
- Implantación larga y difícil

¿Qué es una bbdd?

- “Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados” , (Martin, 1975).
- “Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios” , (Deen, 1985).
- “Conjunto de ficheros maestros, organizados y administrados de una manera flexible de modo que los ficheros puedan ser fácilmente adaptados a nuevas tareas imprevisibles” , (Frank, 1988).



Tipos de datos que se almacenan en una bbdd

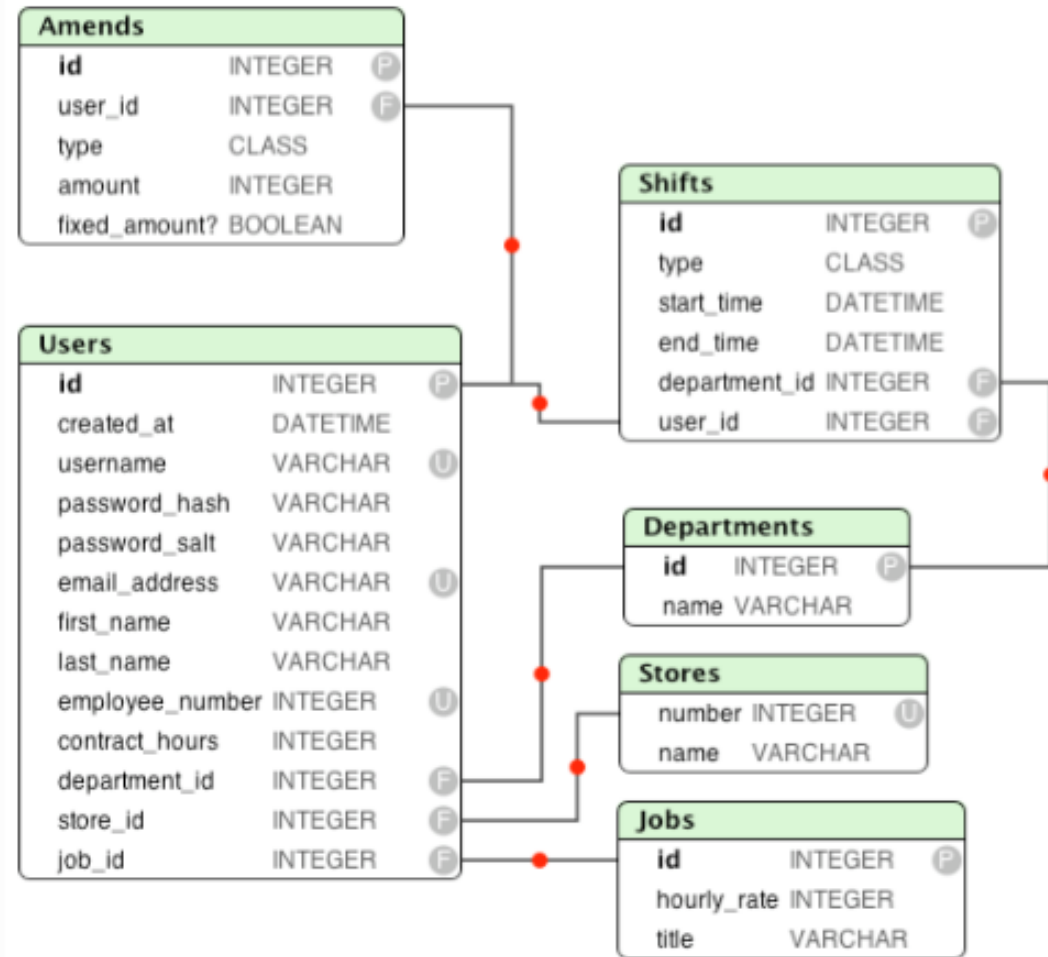


- **Los datos del usuario:** Datos de las aplicaciones de usuarios
Valores de los campos: Nombre, apellidos, codigo_postal,.....
- **Los datos del sistema:** Datos que la base necesita para gestionarse a sí misma
Datos de usuario: nombre_usr, contraseña, privilegios, espacio asignado,...
Datos de archivos: identificación, tamaño, ubicación

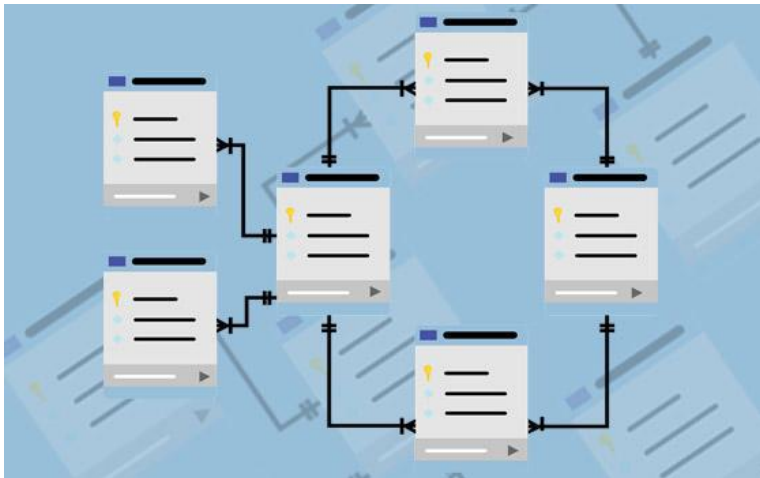
Conceptos asociados a una bbdd



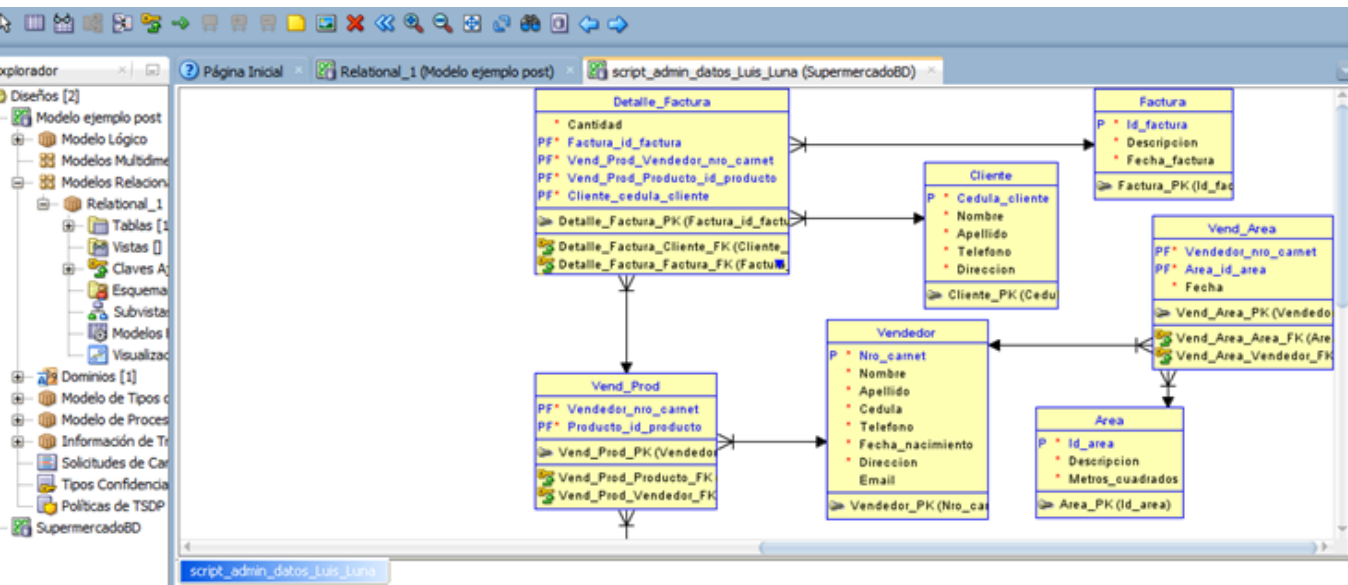
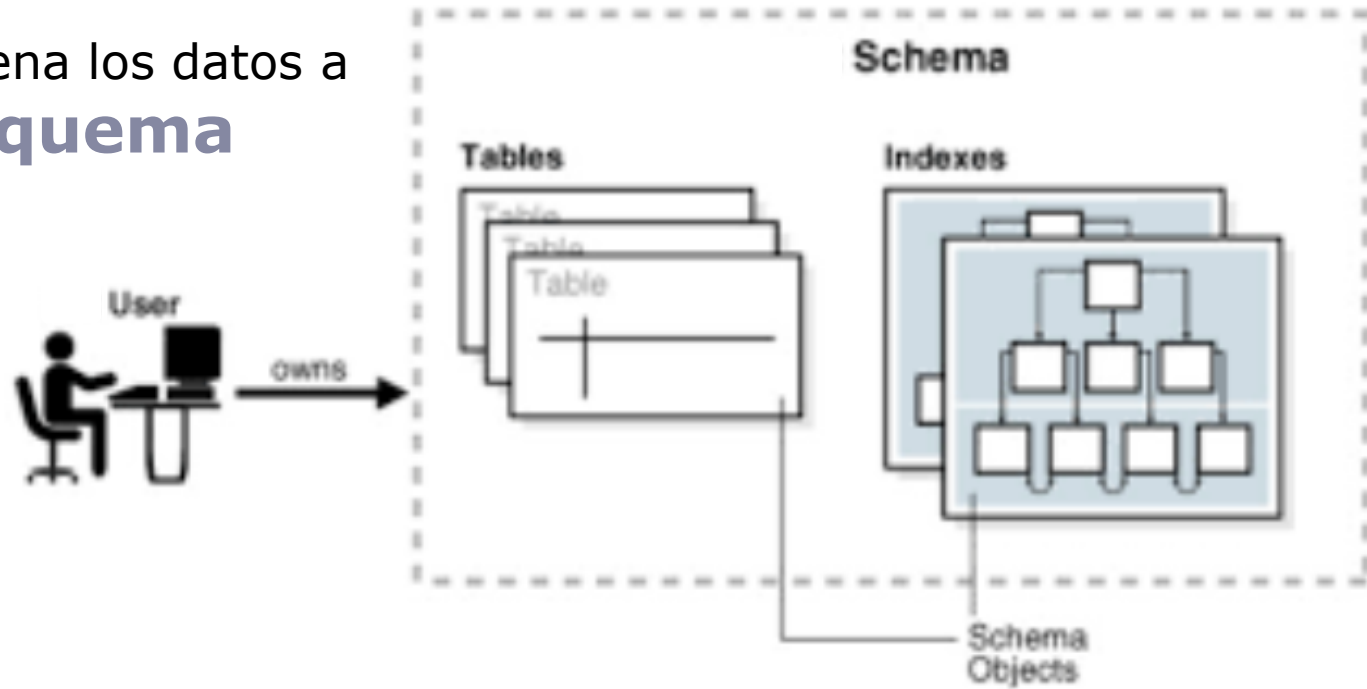
Conceptos asociados a una bbdd



Fuente de la imagen: <https://www.flickr.com/photos/caius/2300154566>



- Una bbdd almacena los datos a través de un **esquema**

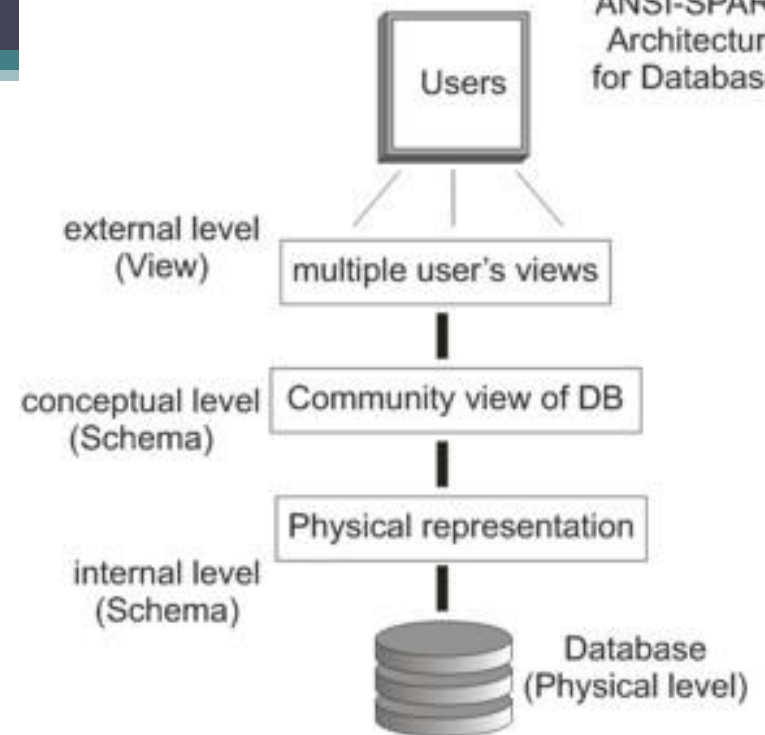
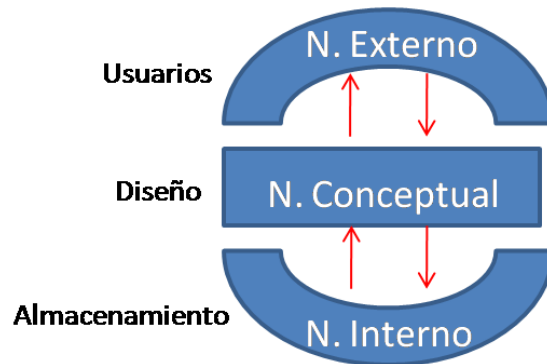


- **char** (n) es una cadena de caracteres completa **character**.
- **varchar** (n) es una cadena de caracteres completa **character varying**.
- **int** es un entero (un subconjunto de los enteros).
- **smallint** es un entero pequeño (un subconjunto de los enteros).
- **numeric** (p,d) es un número en coma decimal con p dígitos, d pertenecen a la parte decimal. Los números 444,5 y 0,32 no se pueden representar con **numeric**.
- **real**, **double precision** son reales de precisión dependiente de la máquina.
- **float** (n) es un número en coma decimal con n dígitos.
- **date** es una fecha del calendario.
- **time** es la hora del día, expresada en minutos y segundos decimales para los segundos (el 1 segundo se divide en 100 partes).
- **timestamp** es una combinación de **date** y **time** (los segundos se dividen en 100 partes).

Un **tipo de datos** es un atributo que especifica el tipo de datos que el objeto puede contener: datos de enteros, datos de caracteres, datos de moneda, datos de fecha y hora, cadenas binarias, etc.

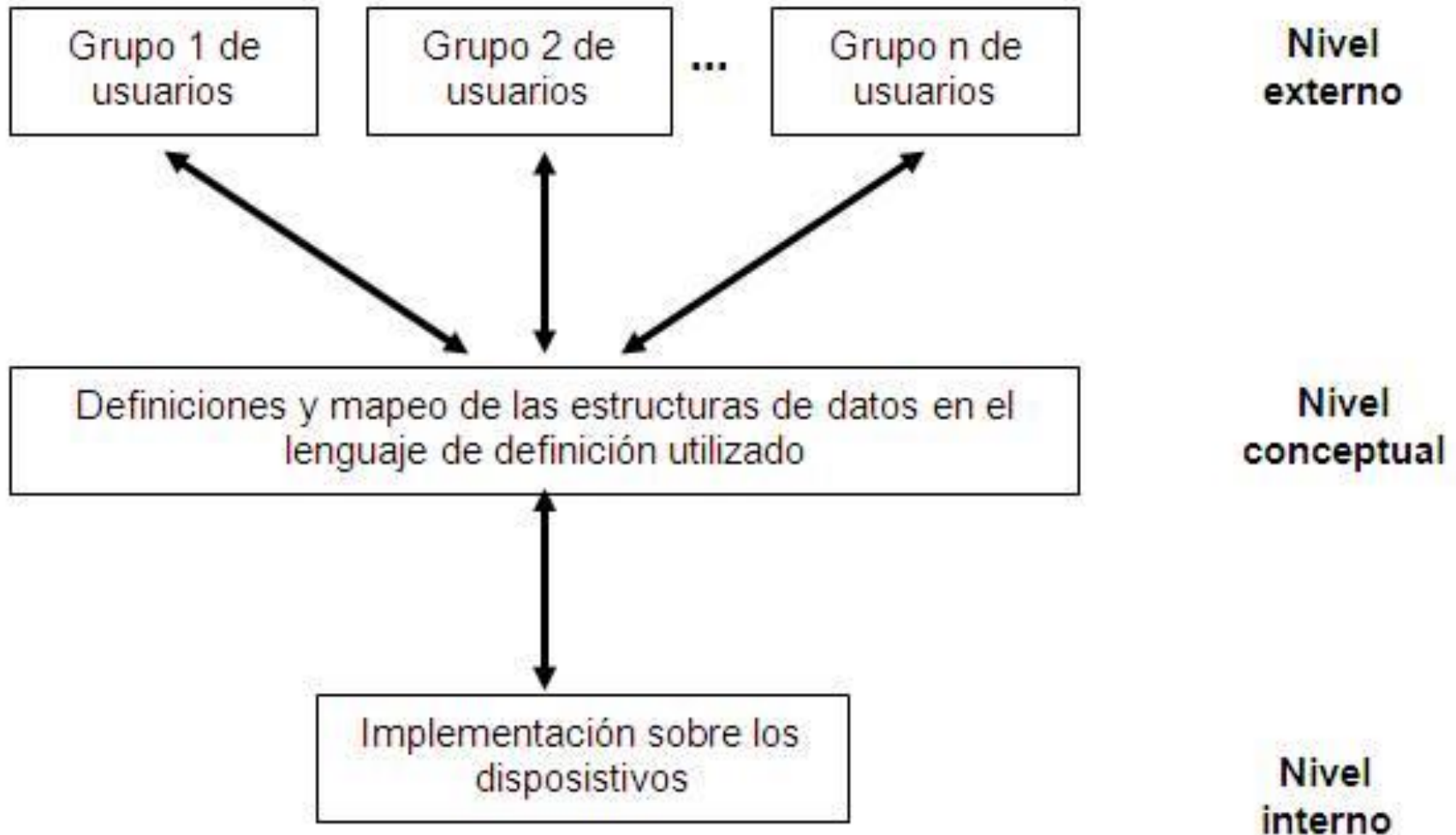
Dependen del **SGBBDD** con el que trabajemos.

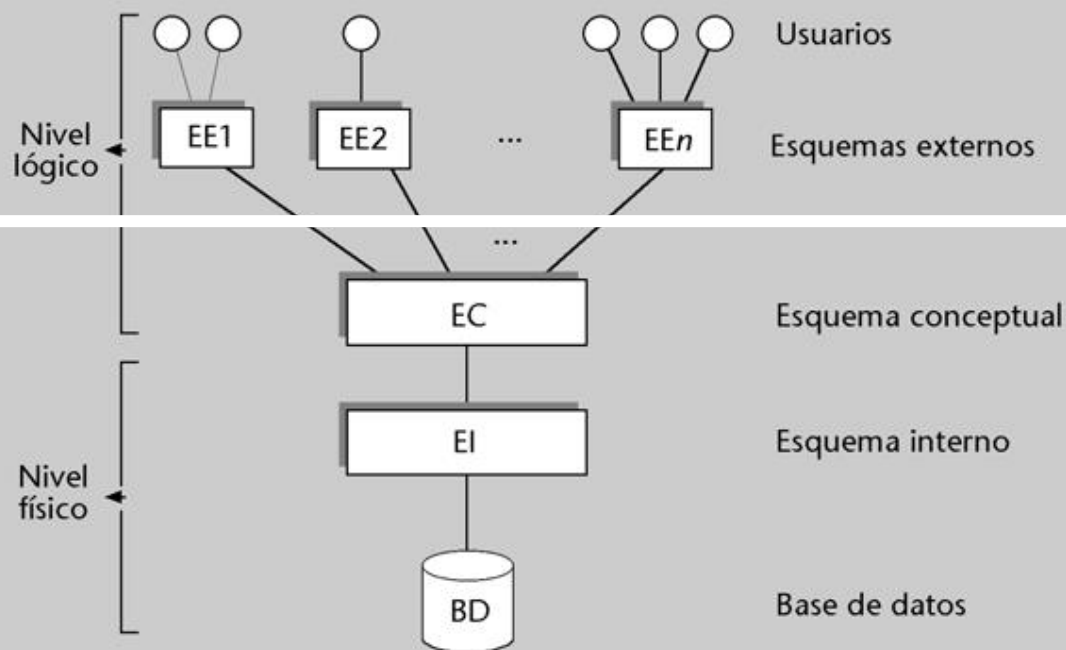
Arquitectura de una bbdd niveles



Estructura de una base de datos Oracle	
Estructura lógica	Vistas
	Tablas
	Tablespaces, índices, clusters, etc.
Estructura física	Ficheros de datos Ficheros de redo log Ficheros de control

Arquitectura de una bbdd niveles





Esquema o nivel Externo: (ejemplo 1)

▫ Esquema1: Departamento ventas

- Consulta ventas del mes

Num_vendedor	nom_vend	producto	cantidad	importe
--------------	----------	----------	----------	---------

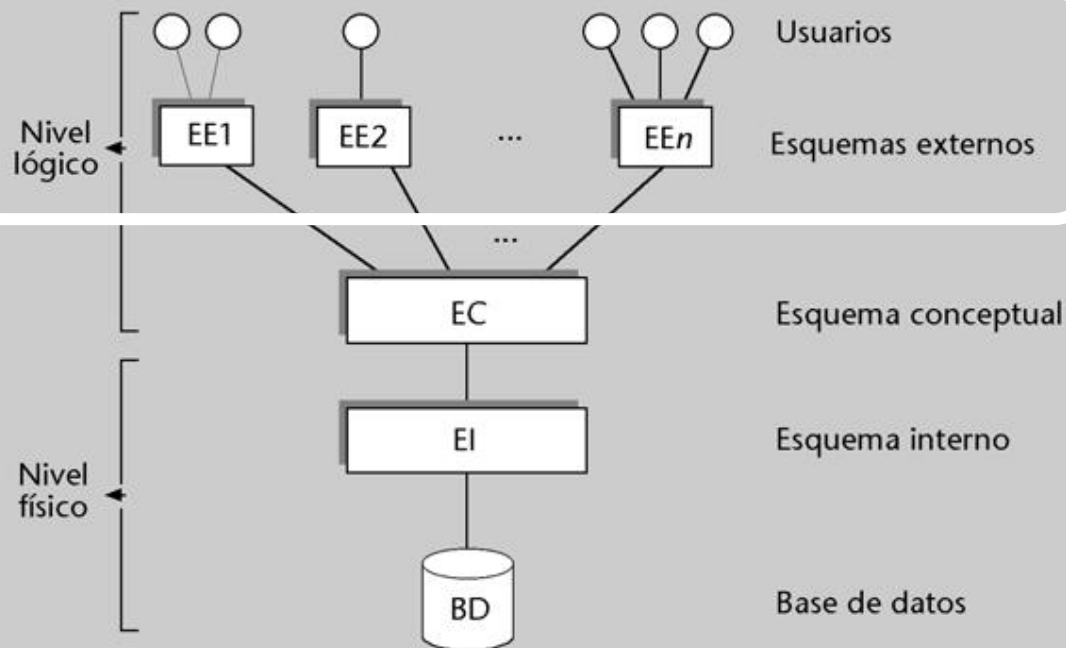
- Consulta de ventas de productos

Cod_producto	nom_prod	cantidad	importe
--------------	----------	----------	---------

▫ Esquema 2: Departamento Almacén

- Consulta de productos en almacén

Cod_producto	nom_prod	cantidad	precio_unidad
--------------	----------	----------	---------------



Esquema o nivel Externo: (ejemplo 2)

Horario laboral de Ajustes

Horario válido desde el 01-05-2012

L	M	X	J	V	S	D
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Botones de control: << Anterior, Actual, Siguiente >>, Nuevo, Copiar horario, Pegar horario, Eliminar Horari, Borrar horas, Excepciones.

Lunes	Martes	Miércoles	Jueves	Viernes
Desde: 16:15 Hasta: 20:30	Desde: 09:30 Hasta: 12:00	Desde: 16:30 Hasta: 20:00	Desde: 09:30 Hasta: 12:15	Desde: Hasta:
Desde: Hasta:	Desde: 12:00 Hasta: 13:15	Desde: 20:15 Hasta: 21:15	Desde: 16:15 Hasta: 20:30	Desde: Hasta:
Desde: Hasta:	Desde: Hasta:	Desde: Hasta:	Desde: Hasta:	Desde: Hasta:

Charlas

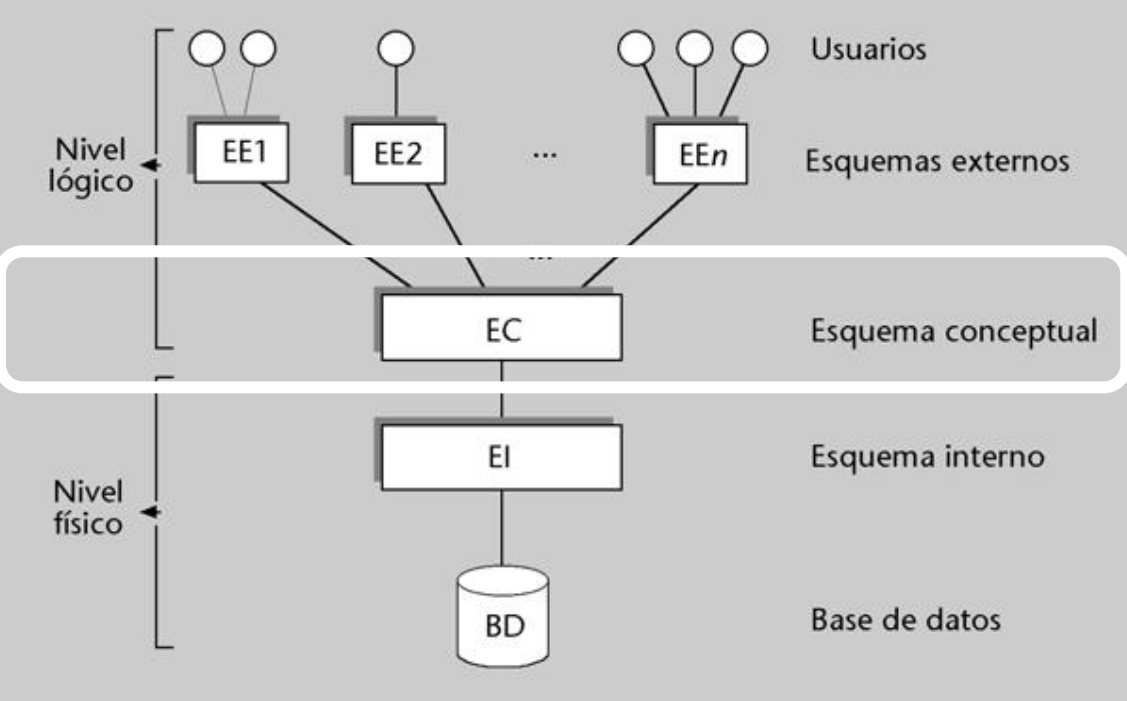
Lunes	Martes	Miércoles	Jueves	Viernes
Desde: Hasta:	Desde: 16:00 Hasta: 17:00	Desde: Hasta:	Desde: Hasta:	Desde: Hasta:
Desde: Hasta:	Desde: Hasta:	Desde: Hasta:	Desde: Hasta:	Desde: Hasta:

Botones: Aceptar, Cancelar

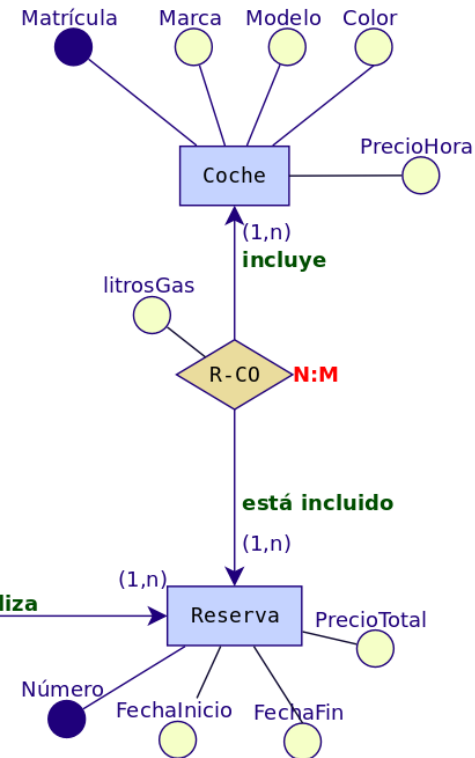
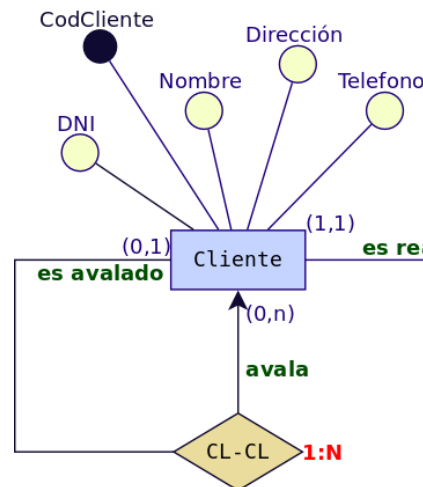
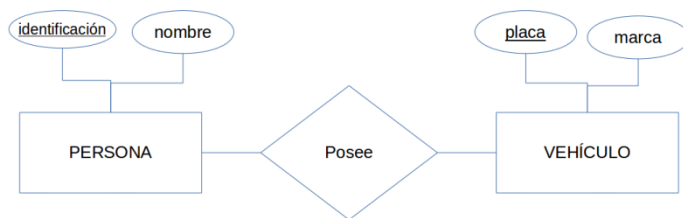
Gestión de horarios personal- Vista 1

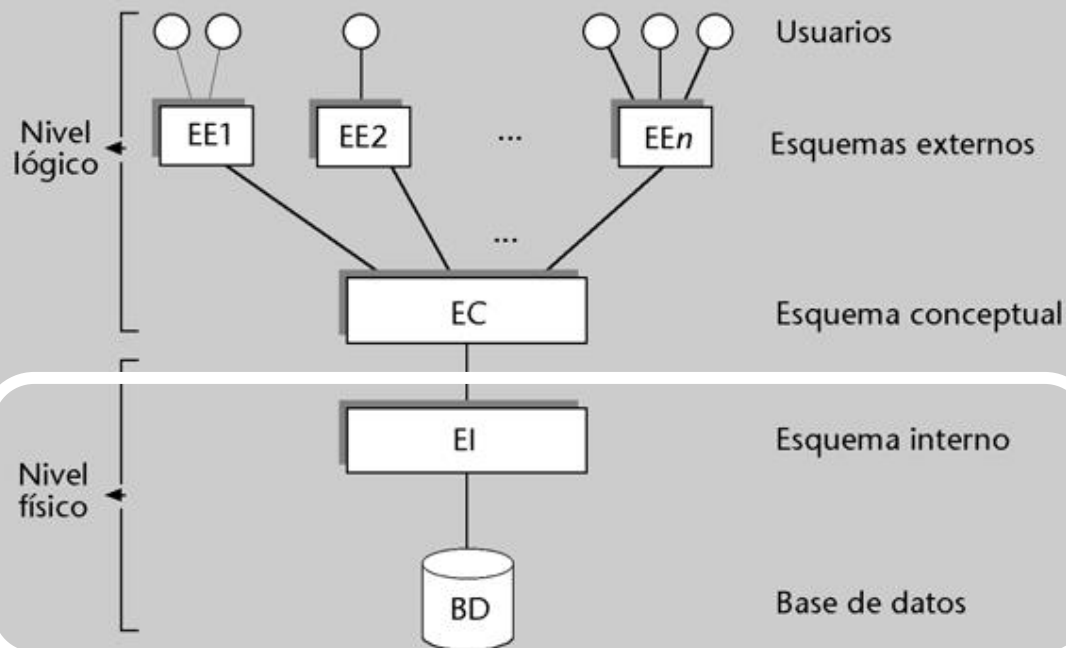
Caja	Resumen	Ausentes	Listado	Horario	Buscar	Marketing	Sa
Hoy							
Septiembre 2012							
L	M	X	J	V	S	D	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
Octubre 2012							
L	M	X	J	V	S	D	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31					
Festivo							

Gestión de citas - Vista 2



Esquema o nivel Conceptual:

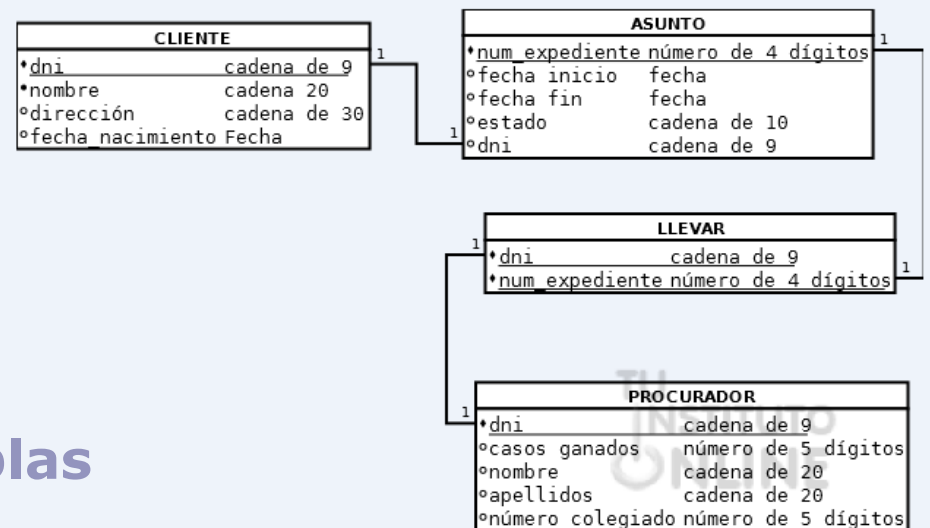




Esquema o nivel Interno:



Trabajamos directamente con los **datos**. Definiremos aquí las **tablas** y reservaremos espacio para los **datos** y los **índices**.



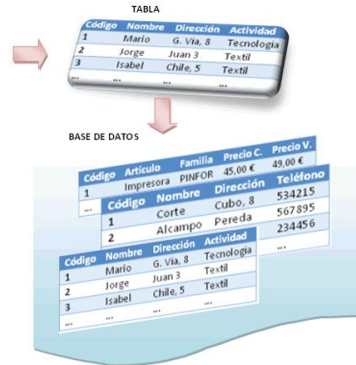
Introducción

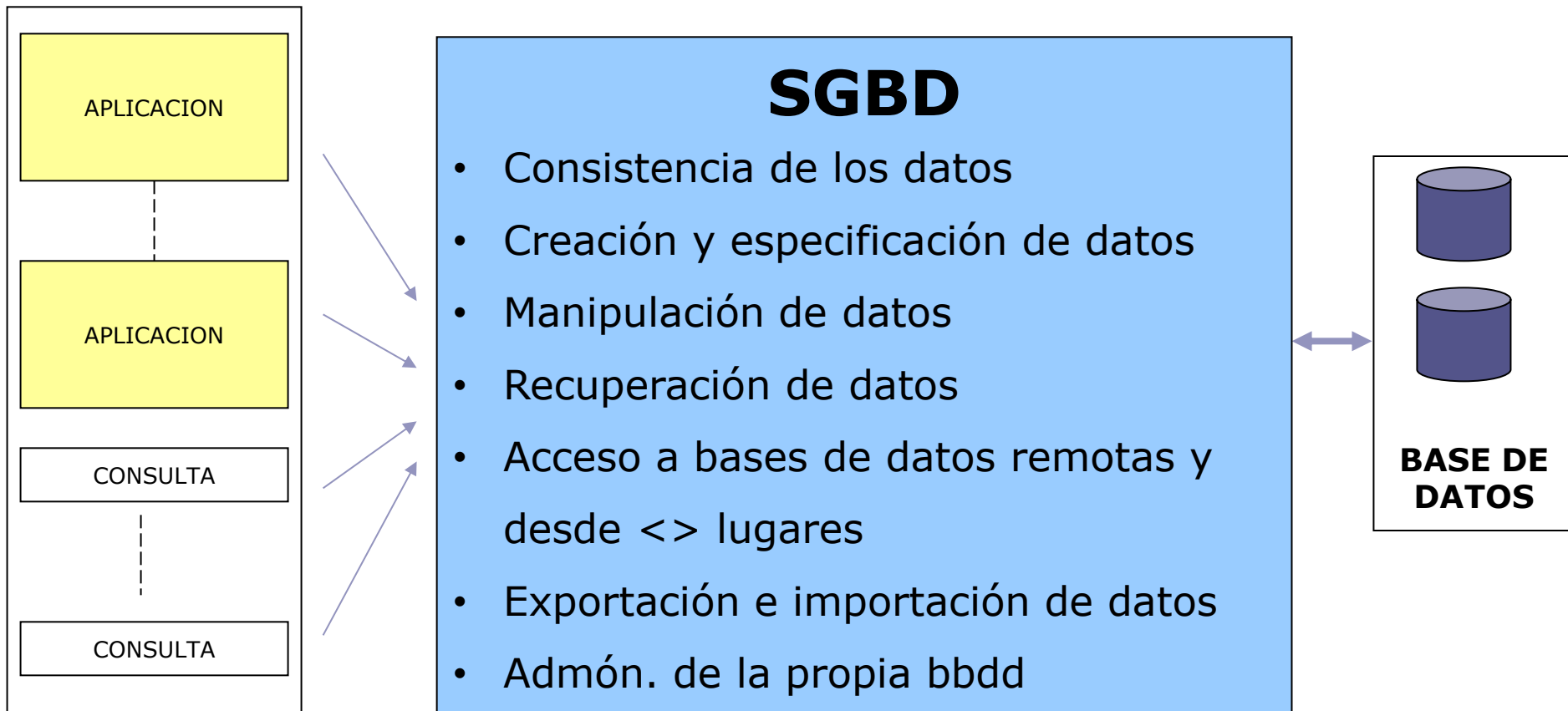
Ficheros

Bases de datos

SGBD. Funciones, componentes y tipos

BBDD centralizadas y BBDD distribuidas



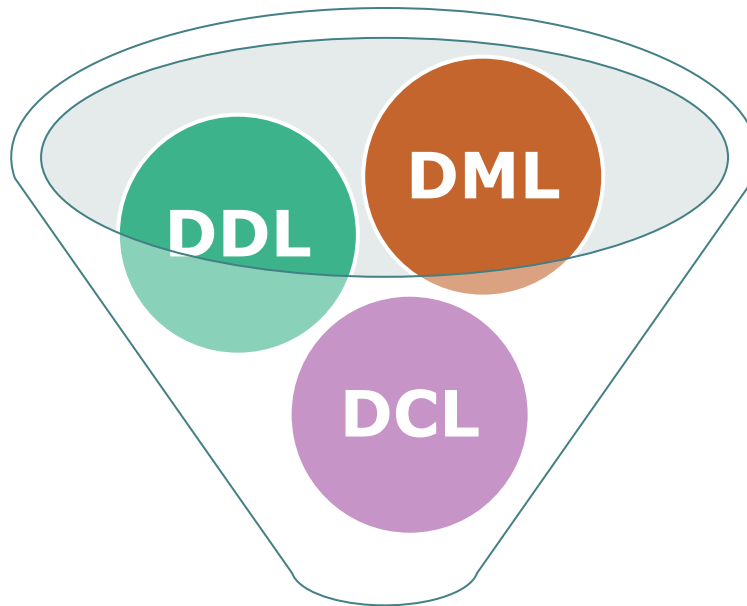


Funciones de un sgbd – Funciones básicas

- Las funciones básicas de un **SGBD** son:
 - Descripción de los datos: **DDL** (Lenguaje Definición de Datos)
 - Manipulación de los datos: **DML** (Lenguaje Manipulación de datos)
 - Transformación de los datos entre los distintos niveles de esq.
externo \leftrightarrow esq. conceptual \leftrightarrow esq. Interno
entre la representación de los datos para el usuario y como se almacenan en el equipo
 - Protección de los datos:
 - Seguridad: tanto lógica como física
 - Integridad
 - Confidencialidad en el acceso
 - Utilidades de administración

Funciones de un sgbd -- Otras funciones

- **Herramientas para la estructura de la base de datos.** Así como para la creación y especificación de los datos.
- **Herramientas para la manipulación de los datos de las bases de datos,** para añadir, modificar, suprimir o consultar datos.
- **Herramientas de recuperación** en caso de desastre
- **Herramientas para la creación de copias de seguridad**
- **Herramientas para la gestión de la comunicación de la base de datos**
- **Herramientas para la creación de aplicaciones** que utilicen esquemas externos de los datos
- **Herramientas de instalación de la base de datos**
- **Herramientas para la exportación e importación de datos**



Descripción \Rightarrow DDL

Manipulación \Rightarrow DML

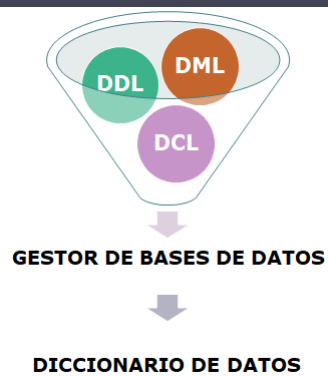
Control \Rightarrow DCL

GESTOR DE BASES DE DATOS

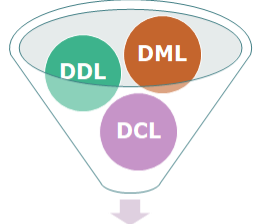


DICCIONARIO DE DATOS

Componentes -- Lenguajes



SELECT	Selección de datos
INSERT UPDATE DELETE MERGE	DML: Lenguaje de manipulación de datos
CREATE ALTER DROP RENAME TRUNCATE	DDL: Lenguaje de definición de datos
COMMIT ROLLBACK SAVEPOINT	Control de transacciones
GRANT REVOKE	DCL: Lenguaje de control de datos



GESTOR DE BASES DE DATOS

DICCIONARIO DE DATOS

Gestor de la bbdd

Software transparente al usuario que garantiza la privacidad, seguridad, integridad y el acceso a los datos, además de la comunicación con el SO.

Interfaz entre los datos, los programas y los usuarios

Diccionario de la bbdd

Donde se encuentra la descripción de toda la bbdd : estructura de los datos, relaciones y la gestión e implementación de la bbdd.

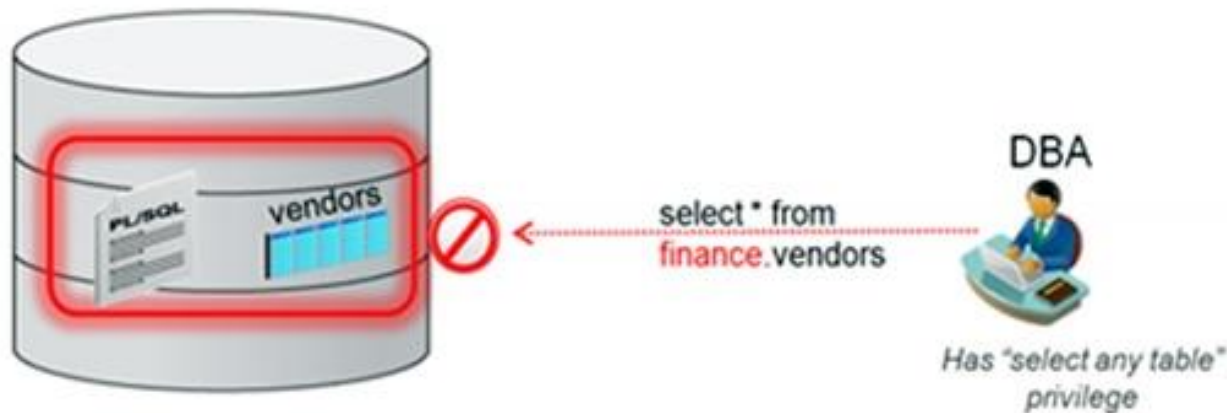
Contendrá:

- Descripción de los distintos niveles de abstracción de la bbdd y sus transformaciones (*no es lo mismo como se representa la información en el ordenador a como la interpreta o ve el usuario*)
- Restricciones de los datos
- Acceso a los datos
- Cuentas de usuario y sus privilegios
- Esquemas externos de cada programa, sus usuarios y autorizaciones

Responsable de la **seguridad**, del **control** y del **mantenimiento** de la bbdd.

Sus tareas:

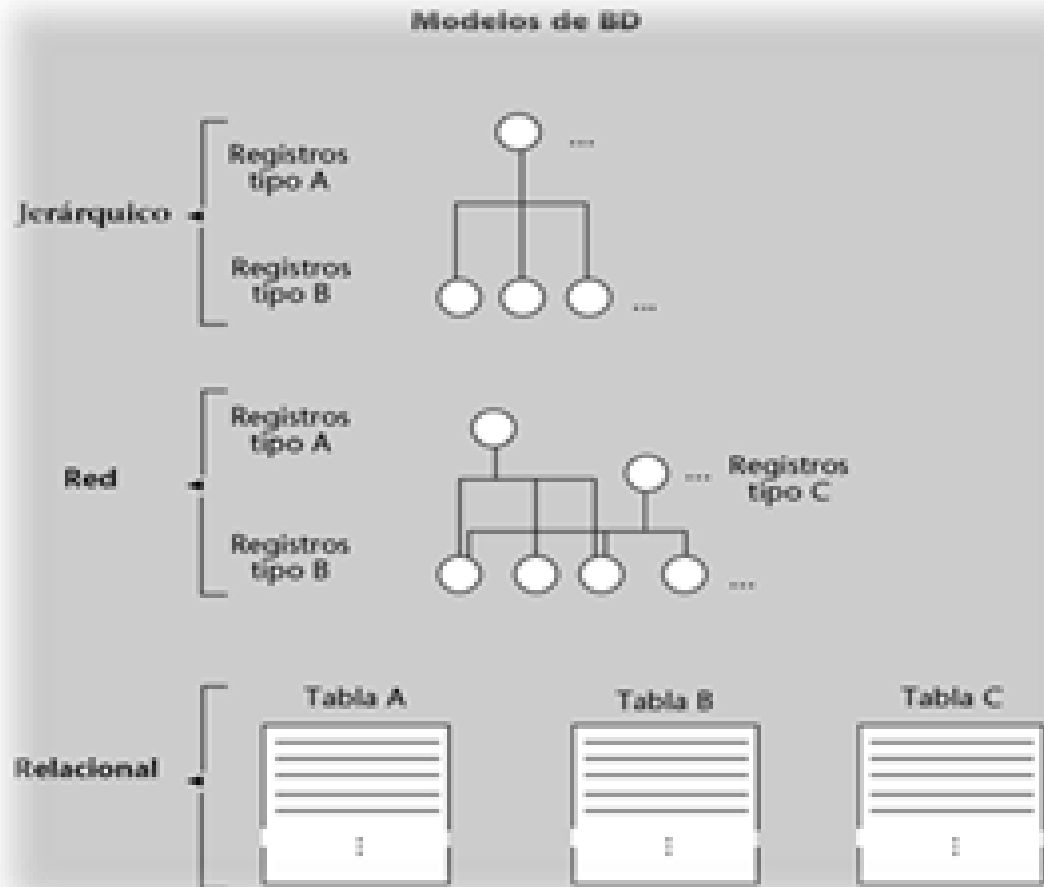
- Definir el esquema lógico
- Definir el esquema físico: estructura almacenamiento y método de acceso
- Subesquemas o visión del usuario de la BD
- Asignación de privilegios
- Seguridad de los datos en caso de fallos
- Mantenimiento del esquema lógico y físico de la BD

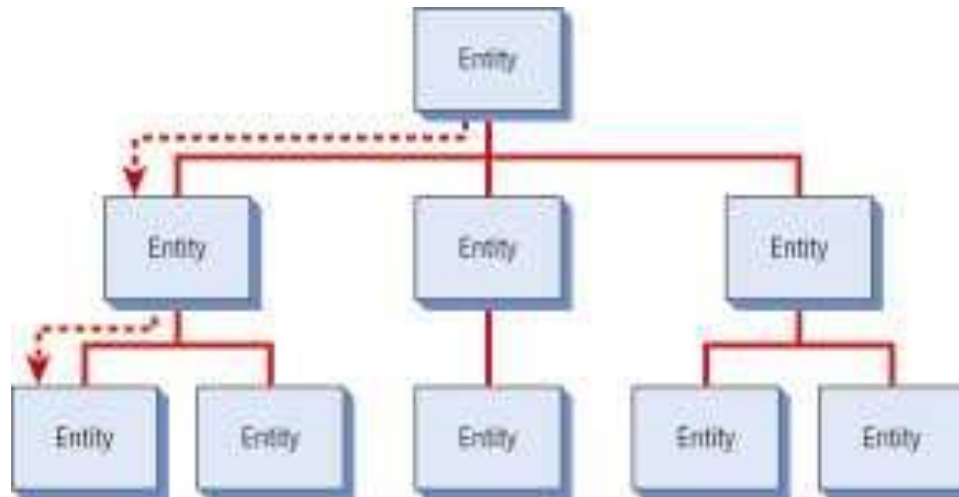


- El **tipo** de un sgbd depende de la bbdd que este implemente.
- En la década de los 70 **DBTG / CODASYL** establece los siguientes **tipos** de bbdd:

**Data
Base
Task
Group**

**Conderence on
Data System
Languages**





Características de este modelo:

Solo existen relaciones 1:M

No se permite más de una relación entre dos segmentos

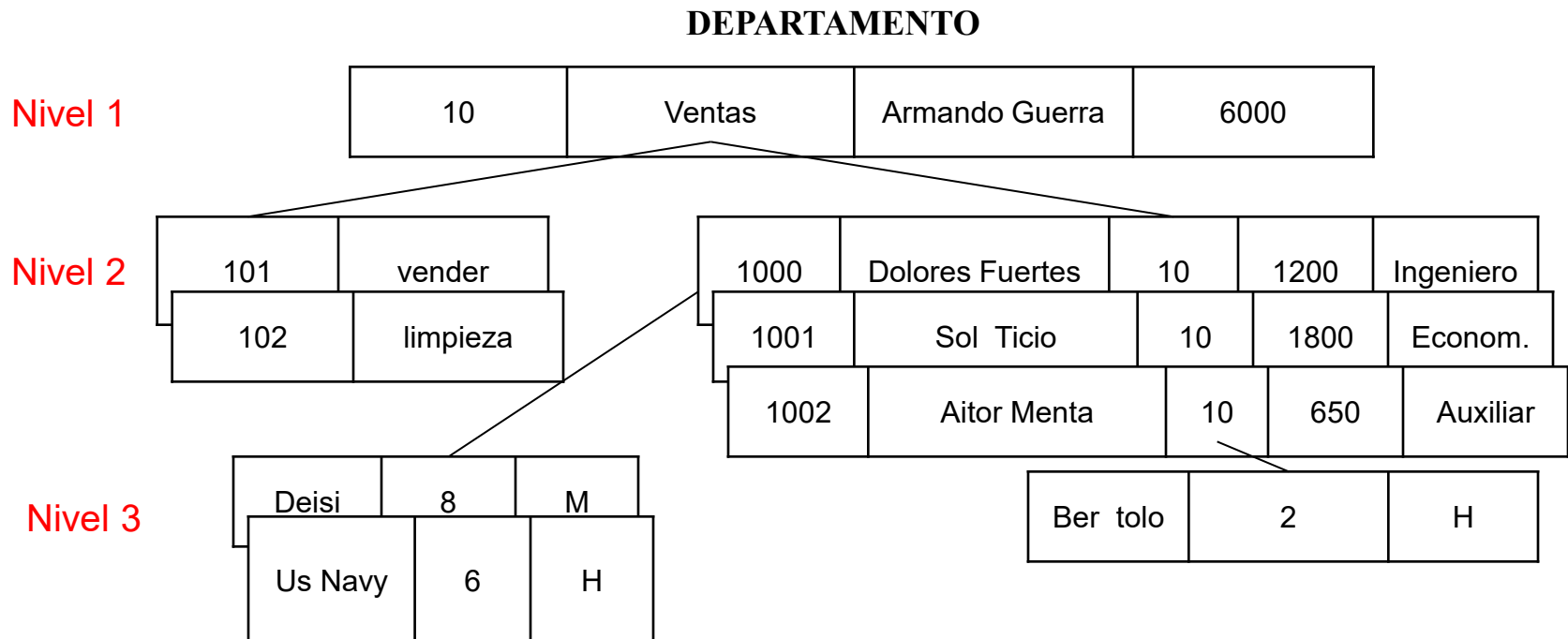
No se admiten relaciones del tipo M:N

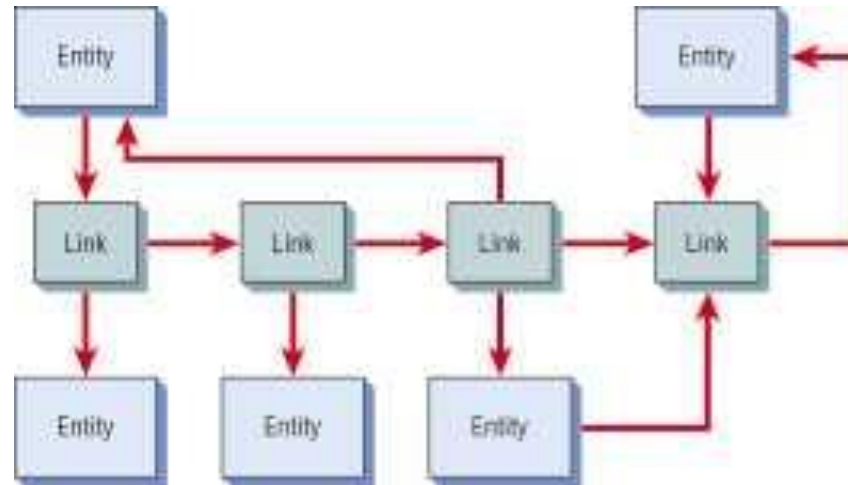
No se permite que un segmento HIJO tenga más de un PADRE

El árbol se recorre en un cierto orden (arriba → abajo)

El acceso a cualquier segmento siempre se realiza a partir del raíz

Modelo jerárquico de bbdd -- ejemplo





Características de este modelo:

Estructura en red

Las entidades o tablas se representan en NODOS

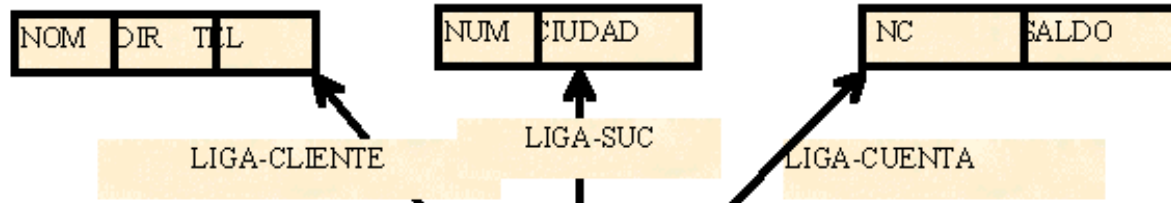
Las relaciones son líneas que unen los nodos

Cualquier entidad se puede vincular con otra (punteros)

Es posible describirla en términos padres e hijos, pero un hijo puede tener varios padres

Modelo en red de bbdd -- ejemplo

CLIENTE
CUENTA



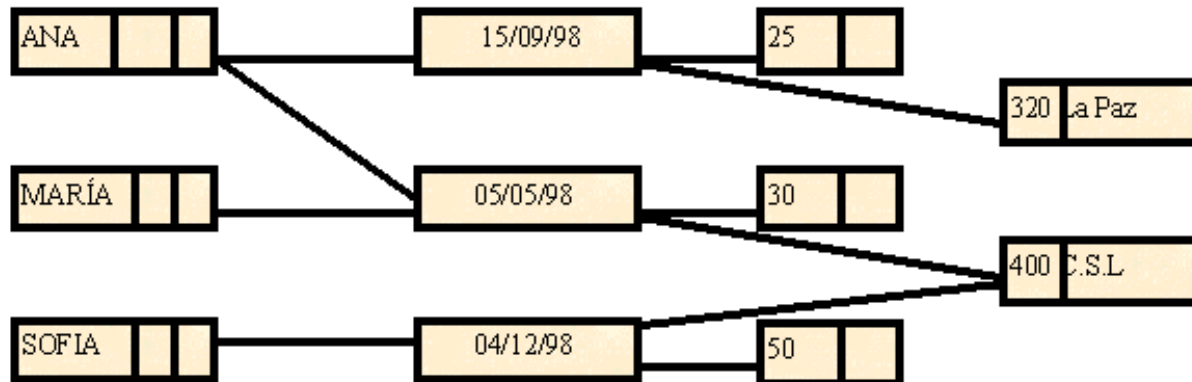
INSTANCIA

CLIENTE

LIGA

CUENTA

SUCURSAL



Tipos de sgbd – Modelo relacional de bbdd

id-cliente	nombre-cliente	calle-cliente	ciudad-cliente
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsa...
32.112.312	Santos	Mayor	Pegu...
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

Tabla cliente

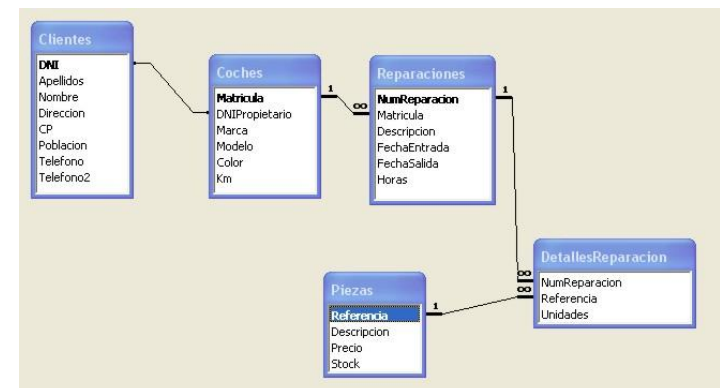
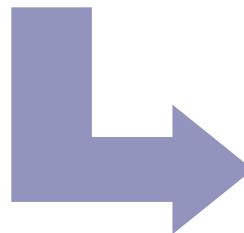
número-cuenta	saldo
C-101	500
C-215	700
C-102	400
C-305	350
C-201	900
C-217	750
C-222	700

Tabla cuenta

id-cliente	número-cuenta
19.283.746	C-101
19.283.746	C-215
01.928.374	C-102
67.789.901	C-305
18.273.609	C-201
32.112.312	C-217
33.666.999	C-222
01.928.374	C-201

Tabla de relación cliente-cuenta

Las **bases de datos relacionales** se caracterizan fundamentalmente porque distribuyen la información en varias tablas que se relacionan entre sí. Con ello se consigue evitar duplicar información y se favorece un acceso más rápido a la misma.



Tipos de sgbd – Modelo relacional de bbdd

Utiliza **tablas bidimensionales** para la representación lógica de los datos y las relaciones

Se apoya en la teoría de **álgebra relacional**

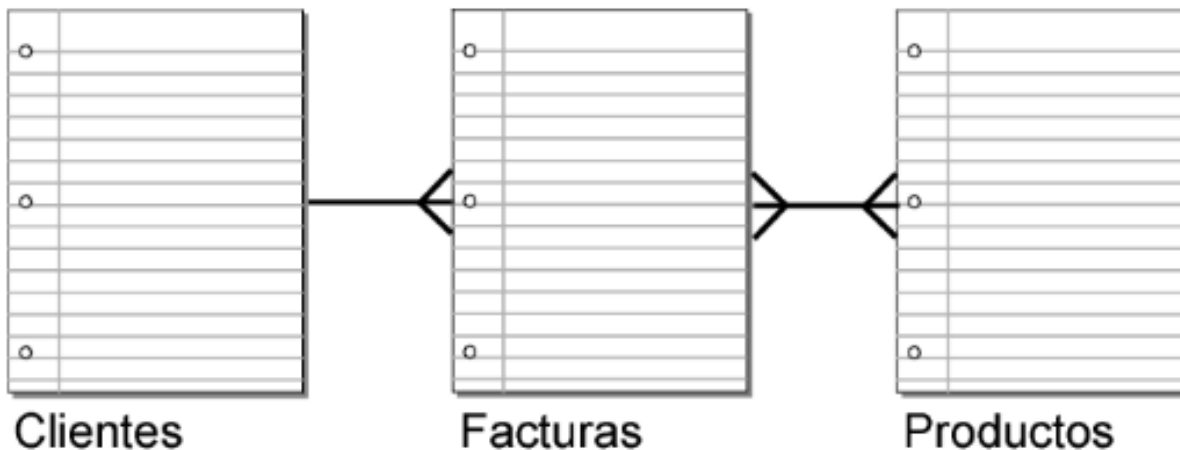
Se trata de un **modelo lógico** que referencia la representación lógica de información y no es directamente aplicable a la representación física

Resumen del modelo:

El elemento principal es la **relación**

Cada relación se representa mediante una **tabla**

Es el modelo más utilizado actualmente



Modelo jerárquico de bbdd -- ejemplo

equipos		
id_equipo	n(3)	pk
nombre	v(50)	nn,uk

jugadores		
id_jugador	n(5)	pk
nombre	v(50)	nn,uk
id_equipo	n(3)	nn,fk

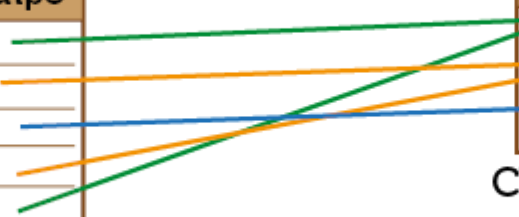


jugadores		
id_jugador	nombre	id_equipo
1	C. Ronaldo	1
2	Messi	2
3	Griezmann	3
4	Luis Suárez	2
5	Bale	1

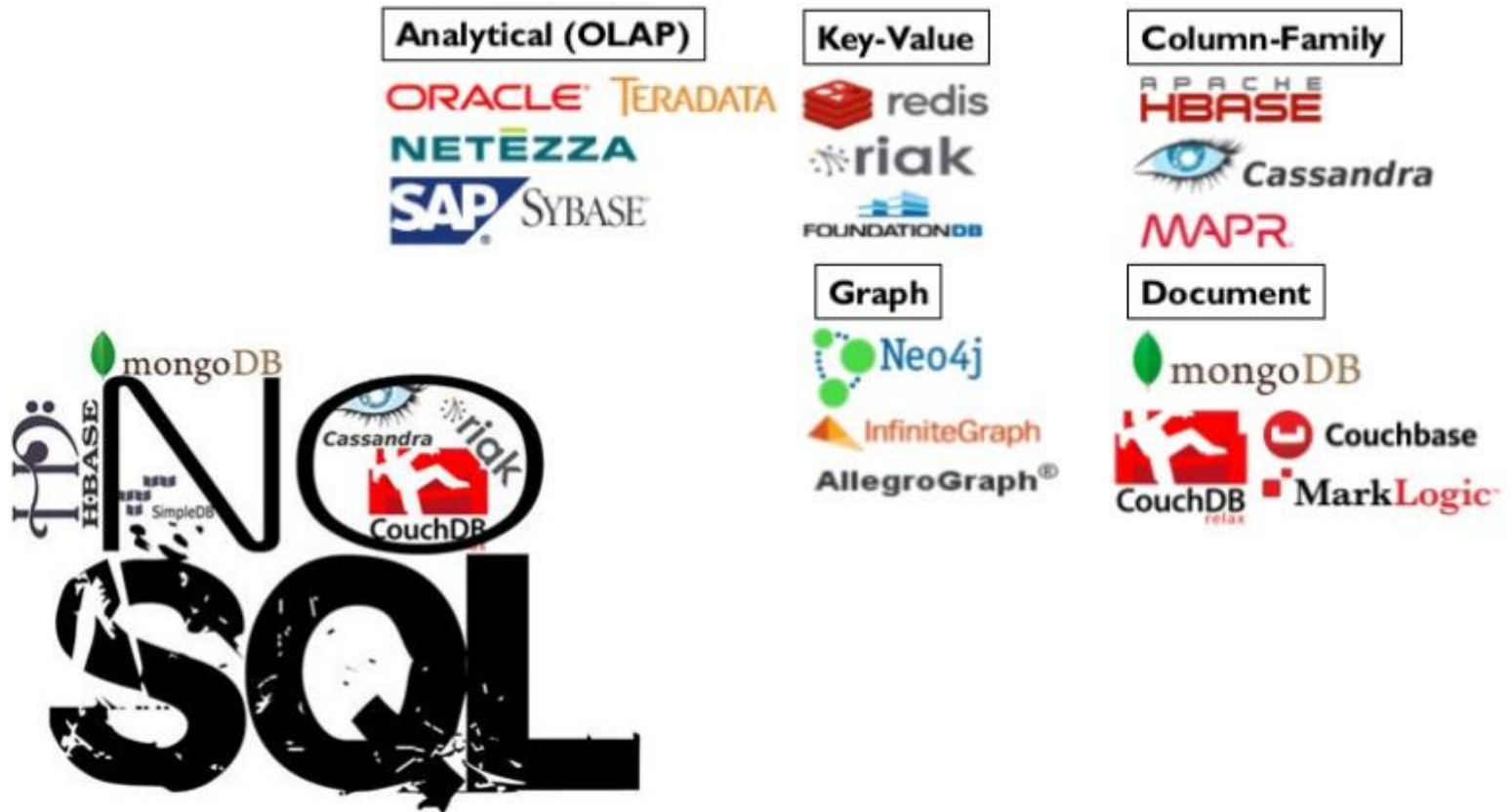
Clave secundaria
FK

equipos	
id_equipo	nombre
1	Real Madrid
2	F.C. Barcelona
3	Atlético de Madrid

Clave primaria
PK



Tipos de sgbd – Otros tipos de bbdd



- Bases de datos clave – valor
- Bases de datos documentales
- Bases de datos en grafo
- Bases de datos en columna
- Bases de datos orientadas a objetos
- ...

Introducción

Ficheros

Bases de datos

SGBD. Funciones, componentes y tipos

BBDD centralizadas y BBDD distribuidas

En función del número de sitios en los que está presente la bbdd

Centralizados sus datos se almacenan en un solo ordenador. Los sgbd centralizados pueden atender a varios usuarios, pero el sgbd y la base de datos en sí residen por completo en una sola máquina.

Distribuidos

la base de datos real y el propio software del sgbd pueden estar distribuidos en **varios sitios** conectados por una red.



The End

Copyright © 2004 by The McGraw-Hill Companies, Inc. All rights reserved.