



Capitulo 2



Conceptos de Sistema y de Arquitectura de Bases de Datos

Temas

2.1 Modelos de datos (Conceptual, Lógico y Físico);
Esquemas, instancias y estado de la BD.

2.2 Arquitectura de 3 niveles. (ANSI/SPARC)

2.3 Independencia de Datos.

2.4 Arquitectura de los SGBD.

2.5 Arquitecturas cliente/servidor y centralizada para los
SGBD

2.6 Breve historia de las bases de datos



2.1 Modelos de Datos

► Definición:

- Un **modelo de datos** es un conjunto de conceptos que describen la **estructura** de una base de datos, las **operaciones** para manipular estas estructuras y algunas **restricciones** que la base de datos debe obedecer.

► Estructura del Modelo de Datos.

- La **estructura** de un modelo se define mediante **construcciones** que incluyen elementos (con su tipo de datos), grupos de elementos (entidades, registros) y relaciones entre estos grupos.

► Restricciones del Modelo de Datos.

- Especifican algunas reglas o validaciones en los datos, y deben de aplicarse todo el tiempo.

► Operaciones del Modelo de Datos.

- Son usadas para obtener y actualizar los datos referidos a las construcciones del modelo. Pueden incluir operaciones básicas (CRUD, Create, Read, Update, Delete) o definidas por el usuario (funciones, procedimientos, vistas).



Categorías de los modelos de datos

- ▶ **Conceptual (de alto nivel, o semántico)**

- ▶ Proporciona conceptos que están más cercanos a cómo los usuarios perciben los datos (también se les conocen como basados en entidades o basados en objetos).

- ▶ **Físico (de bajo nivel, o interno)**

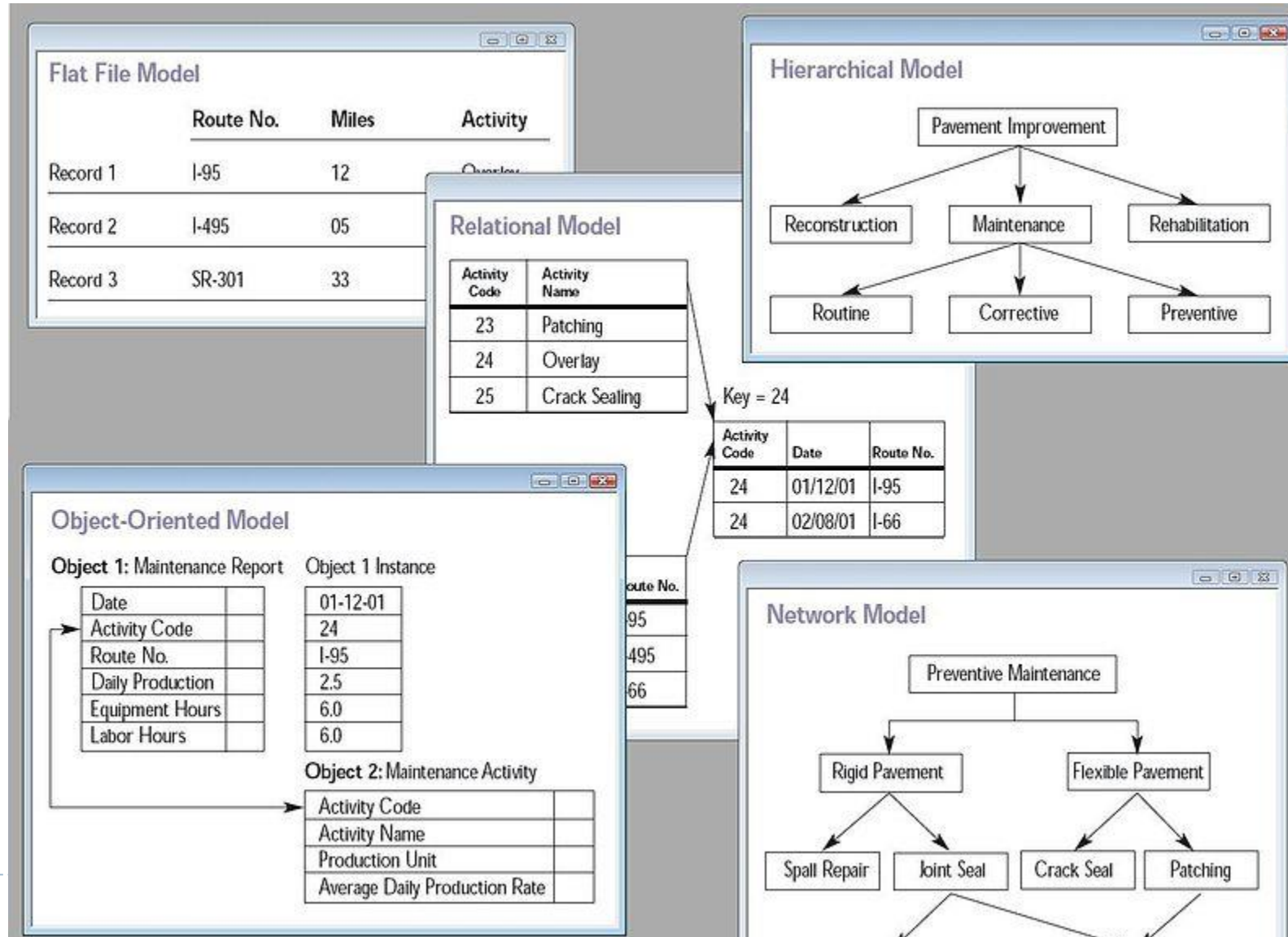
- ▶ Proporciona los conceptos que describen los detalles de cómo los datos son almacenados en la computadora. Son comúnmente especificados en forma concisa por el fabricante del SABD (manuales de administración).

- ▶ **De Implementación (representacional)**

Proporciona los conceptos que están entre los dos mencionados anteriormente, usados por varias implementaciones de SABD comerciales.



Diversos modelos de datos



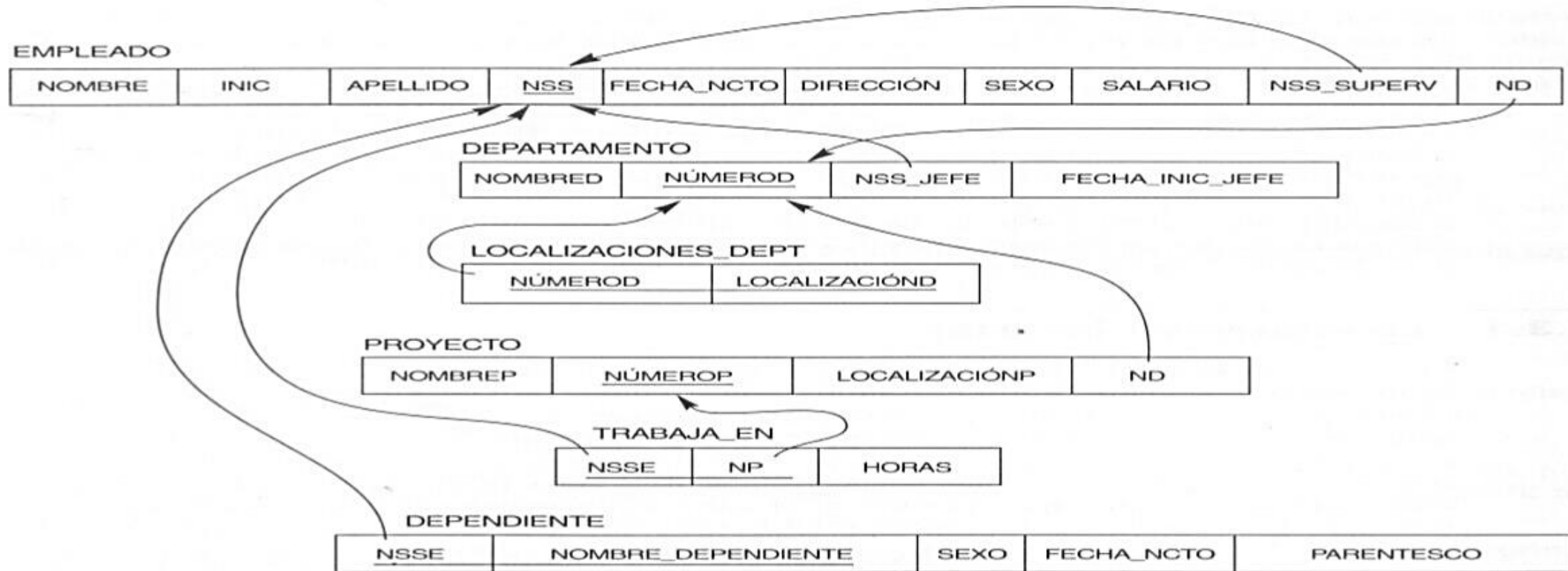
Esquemas e instancias

- ▶ **Esquema de una base de datos.**
 - ▶ Es la descripción de la base de datos.
 - ▶ Incluye la descripción de la estructura de la base de datos, los tipos de datos, y las restricciones de la base de datos.
- ▶ **Diagrama del esquema**
 - ▶ Es una imagen ilustrativa del esquema de la base de datos.
- ▶ **Construcción de un esquema.**
 - ▶ Es un componente del esquema o un objeto dentro del esquema.
- ▶ **Estado de la base de datos.**
 - ▶ Se define como los datos que actualmente están almacenados en la base de datos en un momento dado en el tiempo. Se conoce como una **instancia** de la base de datos.



Esquema de una base de datos

- El **esquema** de una base de datos se refiere como la **intensión** de la base de datos, y se espera de que cambie muy poco en el tiempo.



Estado de una base de datos

- El **estado** de una base de datos se refiere como la **extensión** de la base de datos, y cambia cada vez que la base de datos es actualizada

EMPLEADO	NOMBRE	INIC	APELLIDO	NSS	FECHA_NCTO	DIRECCIÓN	SEXO	SALARIO	NSS_SUPERV	ND
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	H	30000	333445555	5	
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	H	40000	888665555	5	
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	M	25000	987654321	4	
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	M	43000	888665555	4	
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	H	38000	333445555	5	
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	M	25000	333445555	5	
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	H	25000	987654321	4	
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	H	55000	nulo	1	

DEPARTAMENTO	NOMBRED	NUMEROD	NSS_JEFE	FECHA_INIC_JEFE
	Investigación	5	333445555	1988-05-22
	Administración	4	987654321	1995-01-01
	Dirección	1	888665555	1981-06-19

LOCALIZACIONES_DEPT	NUMEROD	LOCALIZACIÓND
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

TRABAJA_EN	NSSE	NP	HORAS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	nulo

PROYECTO	NOMBREP	NUMEROP	LOCALIZACIÓNP	ND
	ProductoX	1	Bellaire	5
	ProductoY	2	Sugarland	5
	ProductoZ	3	Houston	5
	Automatización	10	Stafford	4
	Reorganización	20	Houston	1
	Nuevos beneficios	30	Stafford	4

DEPENDIENTE	NSSE	NOMBRE_DEPENDIENTE	SEXO	FECHA_NCTO	PARENTESCO
	333445555	Alice	M	1986-04-05	HUJA
	333445555	Theodore	H	1983-10-25	HUJO
	333445555	Joy	M	1958-05-03	ESPOSA
	987654321	Abner	H	1942-02-28	ESPOSA
	123456789	Michael	H	1988-01-04	HUJO
	123456789	Alice	M	1988-12-30	HUJA
	123456789	Elizabeth	M	1967-05-05	ESPOSA

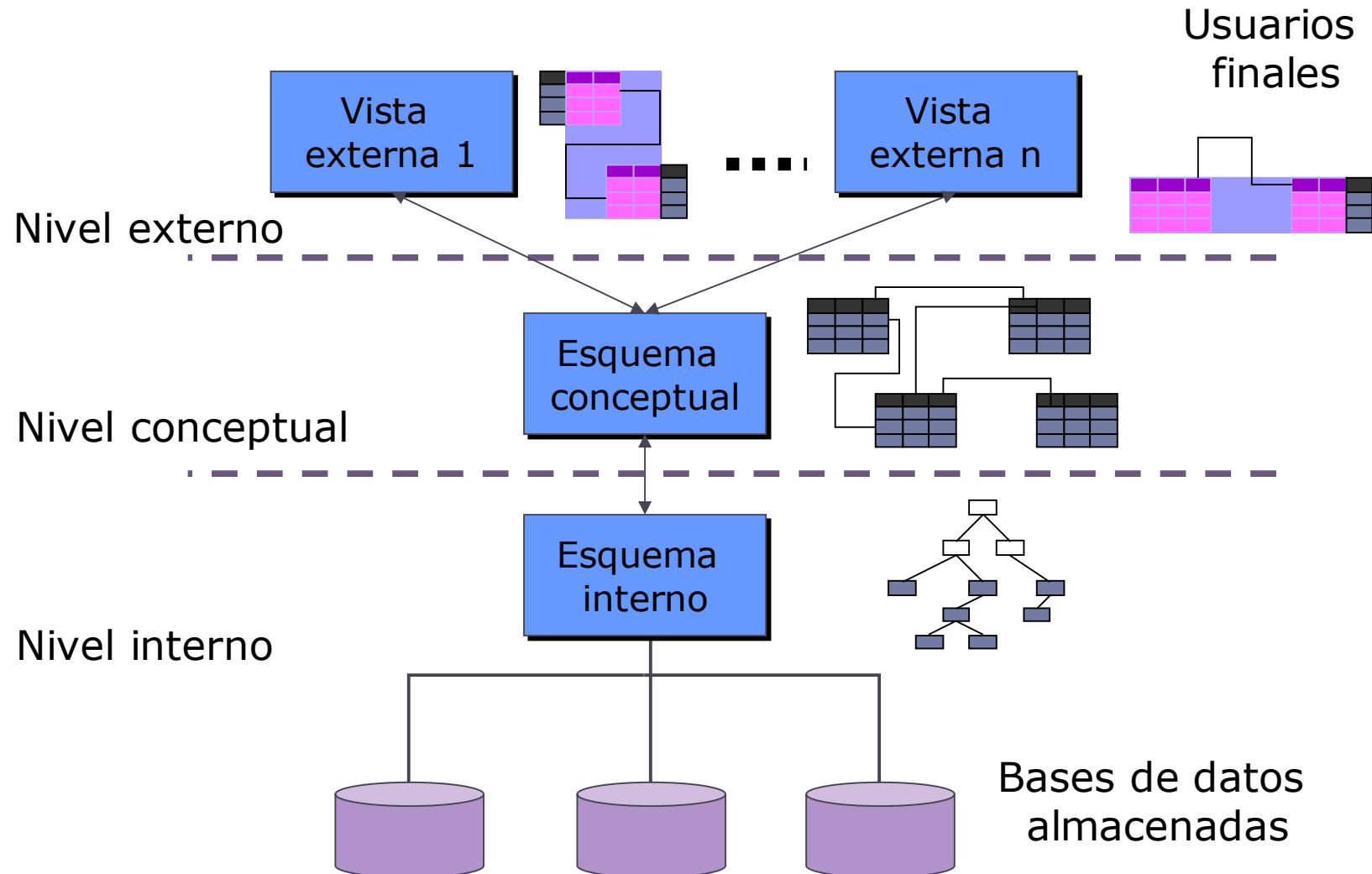
2.2 Arquitectura de tres niveles (ANSI/SPARC)

Definido por la American National Standards Institute / Standards Planning And Requirments Committee en 1975. El modelo que se aplica a una base de datos y contempla varios niveles de esquemas:

- 1.- **El nivel interno**, que describe las estructuras de almacenamiento físico empleadas por la base de datos, y contiene un esquema interno.
- 2.- **El nivel conceptual**, en el que se define la base de datos global para todos los usuarios del sistema, usando un esquema conceptual.
- 3.- **El nivel externo**, en el que cada usuario define su vista de la estructura de la base de datos, ocultando el resto a los demás, usando esquemas externos (vistas de usuario).



Arquitectura de tres niveles (ANSI/SPARC)



2.3 Independencia de datos

- ▶ Se refiere la relación que se tiene de los datos entre las capas de la arquitectura de tres niveles, y se dividen en:
 - ▶ Independencia de datos lógica.
 - ▶ Es la capacidad de cambiar el esquema conceptual sin tener que cambiar los esquemas externos y sus programas de aplicación asociados.
 - ▶ Independencia física de datos.
 - ▶ Es la capacidad de cambiar el esquema interno sin tener que cambiar el esquema conceptual
- ▶ En un SABD que soporte independencia de datos completa, solo deberán cambiar los mapeos entre esquemas de bajo nivel y de niveles superiores. Las aplicaciones no deben verse afectadas por estos cambios.



Lenguajes de un SABD

- ▶ **Lenguaje de Definición de Datos (Data Definition Language)**
 - ▶ Es usado por el administrador (DBA) y los diseñadores de la base de datos para especificar el esquema conceptual.
 - ▶ También permite, en muchos SABD, definir los esquemas internos y externos (vistas).
 - ▶ En otros SABD, están separados como el Lenguaje de Definición de Almacenamiento (Storage Definition Language) y el Lenguaje de Definición de Vistas (View Definition Language).



Lenguajes de un SABD

► Lenguaje de Manipulación de Datos.

- Es usado para especificar consultas y actualizaciones al contenido de la base de datos.
- Se consideran dos formas de implementación:
 - Lenguajes de alto nivel o no procedurales.- Se orientan a especificar que datos obtener más que en cómo obtenerlos. Son llamados lenguajes declarativos (ej. el Lenguaje Estructurado de Consultas - SQL)
 - Lenguajes de bajo nivel o procedurales.- Obtienen los datos un registro a la vez, por lo que necesitan estructuras de control mediante un lenguaje de programación (ej. SQL incrustado).

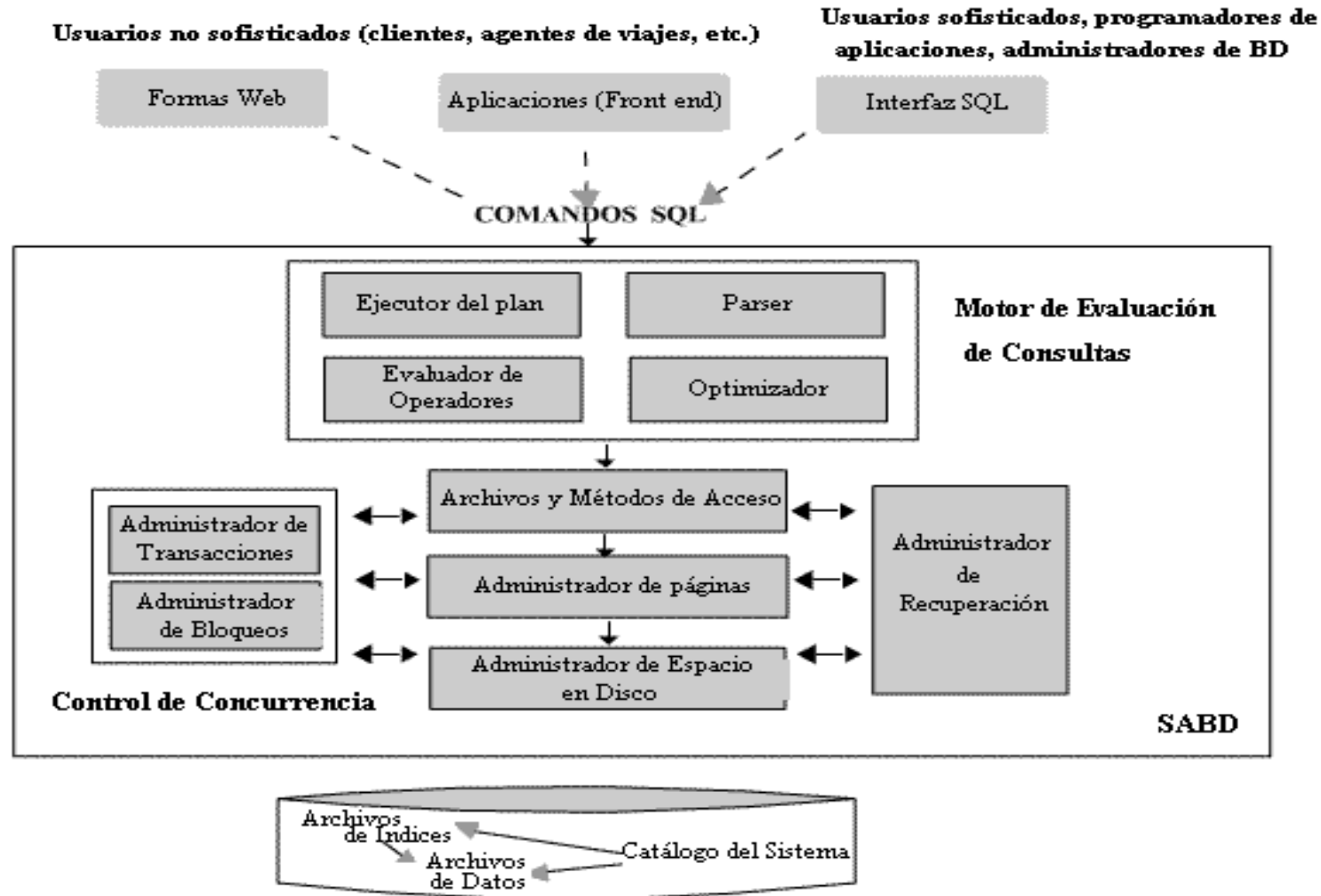


Utilerías de un SABD

- ▶ Para la administración de un sistema de bases de datos, generalmente se han desarrollado aplicaciones de software que facilitan su uso, tales como:
 - ▶ Carga de datos almacenados en archivos (importación).
 - ▶ Respaldo de la base de datos de forma periódica en un medio de almacenamiento físico.
 - ▶ Reorganización de la estructura de la base de datos.
 - ▶ Generación de reportes.
 - ▶ Monitoreo del rendimiento del sistema.
 - ▶ Otras más, como análisis de estadísticas, seguridad, etc.

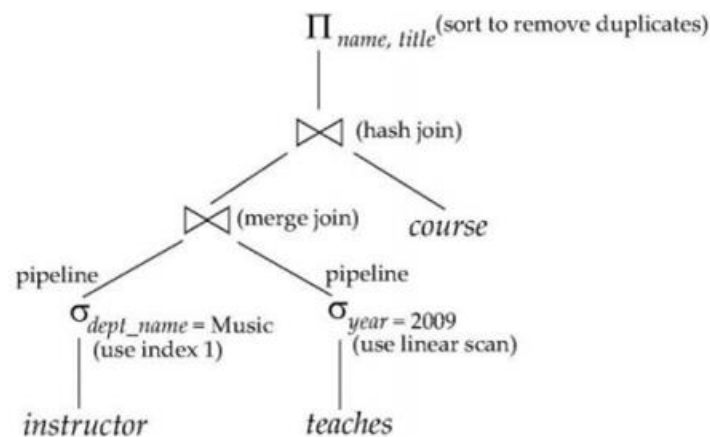


2.4.- Arquitectura de un SABD



Procesamiento de consultas (1)

- ▶ Cuando un usuario envía una consulta, el módulo **compilador** de SQL la descompone y analiza, para posteriormente pasarla al **optimizador**, quien encuentra el **plan de ejecución** más eficiente para evaluar la consulta, empleando técnicas basadas en reglas y en costos.
- ▶ El plan de ejecución se presenta comúnmente como un árbol de operadores relacionales, el cual se almacena en el SABD para usarse posteriormente, en caso de que se ejecute nuevamente la misma consulta.



```
SQL_ID 47ju6l02uvq5q, child number 0
-----
SELECT e.last_name, d.department_name
FROM hr.employees e, hr.departments d WHERE
e.department_id = d.department_id

Plan hash value: 2933537672
-----
```

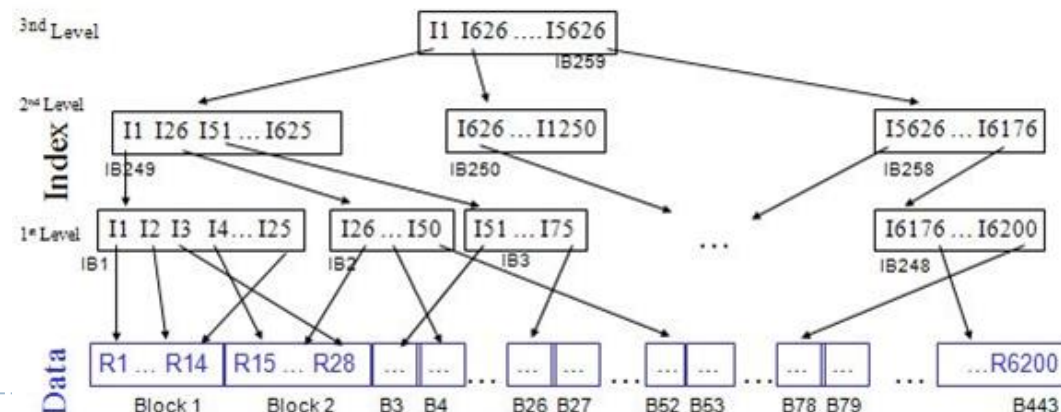
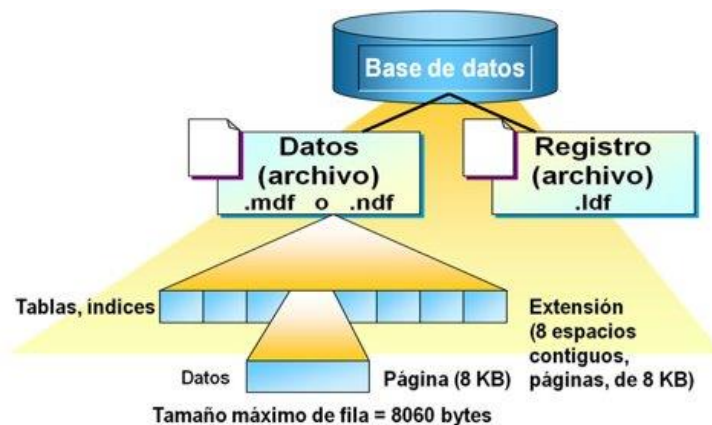
Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT				6 (100)
1	MERGE JOIN		106	2862	6 (17)
2	TABLE ACCESS BY INDEX ROWID	DEPARTMENTS	27	432	2 (0)
3	INDEX FULL SCAN	DEPT_ID_PK	27		1 (0)
* 4	SORT JOIN		107	1177	4 (25)
5	TABLE ACCESS FULL	EMPLOYEES	107	1177	3 (0)

```
Predicate Information (identified by operation id):
-----
4 - access("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")
   filter("E"."DEPARTMENT_ID"="D"."DEPARTMENT_ID")

24 rows selected.
```

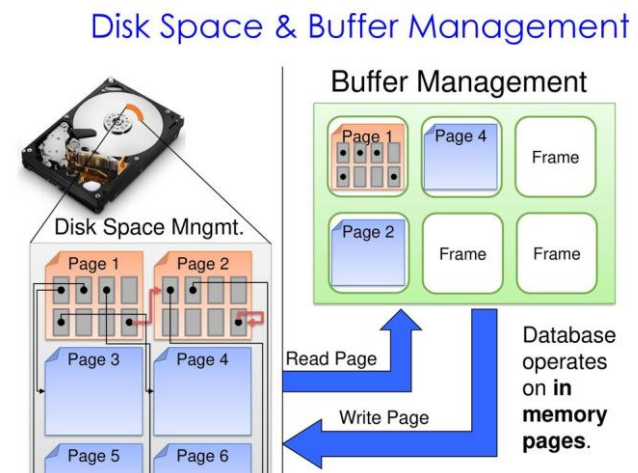
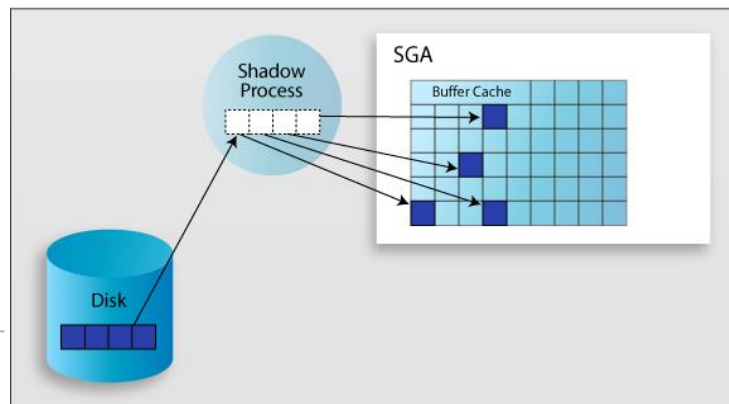

Procesamiento de consultas (2)

- ▶ La **capa de archivos y métodos de acceso** incluye una variedad de software para soportar el concepto de archivo como una colección de páginas o de registros, el cual hace uso intensivo del **catálogo del sistema** (metadatos) para hacer el mapeo del nivel conceptual al nivel interno.
- ▶ Los métodos de acceso son los **índices** creados para las tablas, comúnmente empleando algoritmos de búsqueda como los **árboles B+** o técnicas **Hash**. Esto permite acelerar la búsqueda de datos indexados.



Procesamiento de consultas (3)

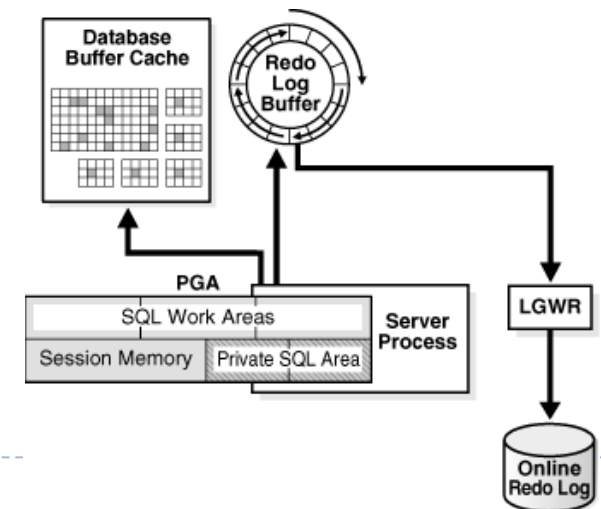
- ▶ Los **archivos y métodos de acceso** están apoyados por el **administrador de páginas** (buffer), el cual transfiere y desaloja páginas de datos de memoria hacia el disco físico, según la demanda. Tiene **algoritmos de reemplazo** de páginas para hacer más eficiente el rendimiento del sistema.
- ▶ El nivel más bajo corresponde al **administrador de espacio en disco**, en donde los datos están almacenados. Se encarga de emplear rutinas de lectura y escritura de sectores físicos del disco, generalmente mediante el sistema de archivos del SO.



Procesamiento de consultas (4)

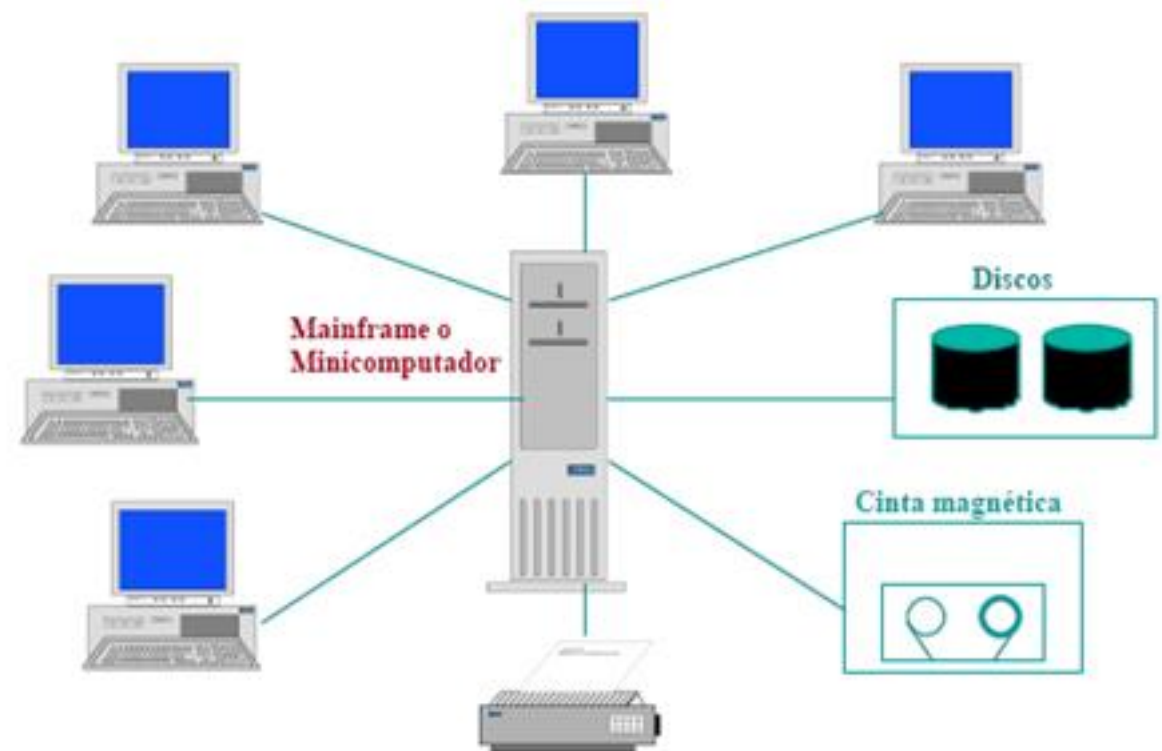
- ▶ El SABD soporta la **conurrencia** multiusuario y la **recuperación de fallos** por medio de calendarizaciones y de registros en **bitácoras** de los cambios en la base de datos.
- ▶ El **administrador de transacciones** se encarga del control de concurrencia, que habilita que los usuarios no interfieran entre sí.
- ▶ El **administrador de bloqueos** lleva un control de los recursos bloqueados por los usuarios y de la liberación de estos para ser usados posteriormente.
- ▶ El **administrador de recuperación** es responsable de restaurar la consistencia de la base de datos posterior a una falla del sistema

Results		Messages								
	Current LSN	Operation	Transaction ID	Parent Transaction ID	Begin Time	Transaction Name	Transaction SID			
1	00000020:00000055:0001	LOP_BEGIN_XACT	0000:000002cf	NULL	2013/05/06 12:54:04:977	Backup/CommitDifferentialBase	0x010500000000000515<			
2	00000020:0000005d:0002	LOP_BEGIN_XACT	0000:000002d0	NULL	2013/05/06 12:54:05:037	CREATE TABLE	0x010500000000000515<			
3	00000020:0000005d:0003	LOP_BEGIN_XACT	0000:000002d1	0000:000002d0	2013/05/06 12:54:05:037	CMEDCatYukonObject::PreAllocateO	0x010500000000000515<			
4	00000020:0000005f:000e	LOP_BEGIN_XACT	0000:000002d2	0000:000002d0	2013/05/06 12:54:05:050	SplitPage	0x010500000000000515<			
5	00000020:0000005f:003f	LOP_BEGIN_XACT	0000:000002d3	0000:000002d0	2013/05/06 12:54:05:050	SplitPage	0x010500000000000515<			
6	00000020:0000005f:0058	LOP_BEGIN_XACT	0000:000002d4	0000:000002d0	2013/05/06 12:54:05:053	SplitPage	0x010500000000000515<			
7	00000020:0000005f:0074	LOP_BEGIN_XACT	0000:000002d5	0000:000002d0	2013/05/06 12:54:05:053	SplitPage	0x010500000000000515<			
8	00000020:0000005b:0002	LOP_BEGIN_XACT	0000:000002d6	NULL	2013/05/06 12:54:05:070	INSERT	0x010500000000000515<			
9	00000020:0000005b:0004	LOP_BEGIN_XACT	0000:000002d7	NULL	2013/05/06 12:54:05:070	Allocate Root	0x010500000000000515<			
10	00000020:0000005b:0006	LOP_BEGIN_XACT	0000:000002d8	0000:000002d7	2013/05/06 12:54:05:070	AllocFirstPage	0x010500000000000515<			
11	00000020:000000a0:0002	LOP_BEGIN_XACT	0000:000002d9	NULL	2013/05/06 12:54:05:073	INSERT	0x010500000000000515<			
12	00000020:000000a1:0001	LOP_BEGIN_XACT	0000:000002da	NULL	2013/05/06 12:54:05:077	INSERT	0x010500000000000515<			
13	00000020:000000a2:0001	LOP_BEGIN_XACT	0000:000002db	NULL	2013/05/06 12:54:05:077	INSERT	0x010500000000000515<			
14	00000020:000000a3:0001	LOP_BEGIN_XACT	0000:000002dc	NULL	2013/05/06 12:54:05:080	INSERT	0x010500000000000515<			
15	00000020:000000a4:0001	LOP_BEGIN_XACT	0000:000002dd	NULL	2013/05/06 12:54:05:090	user_transaction	0x010500000000000515<			
16	00000020:000000a7:0001	LOP_BEGIN_XACT	0000:000002de	NULL	2013/05/06 12:54:05:107	UpdateQPStats	0x010500000000000515<			
17	00000020:000000aa:0001	LOP_BEGIN_XACT	0000:000002df	NULL	2013/05/06 12:54:05:113	DELETE	0x010500000000000515<			



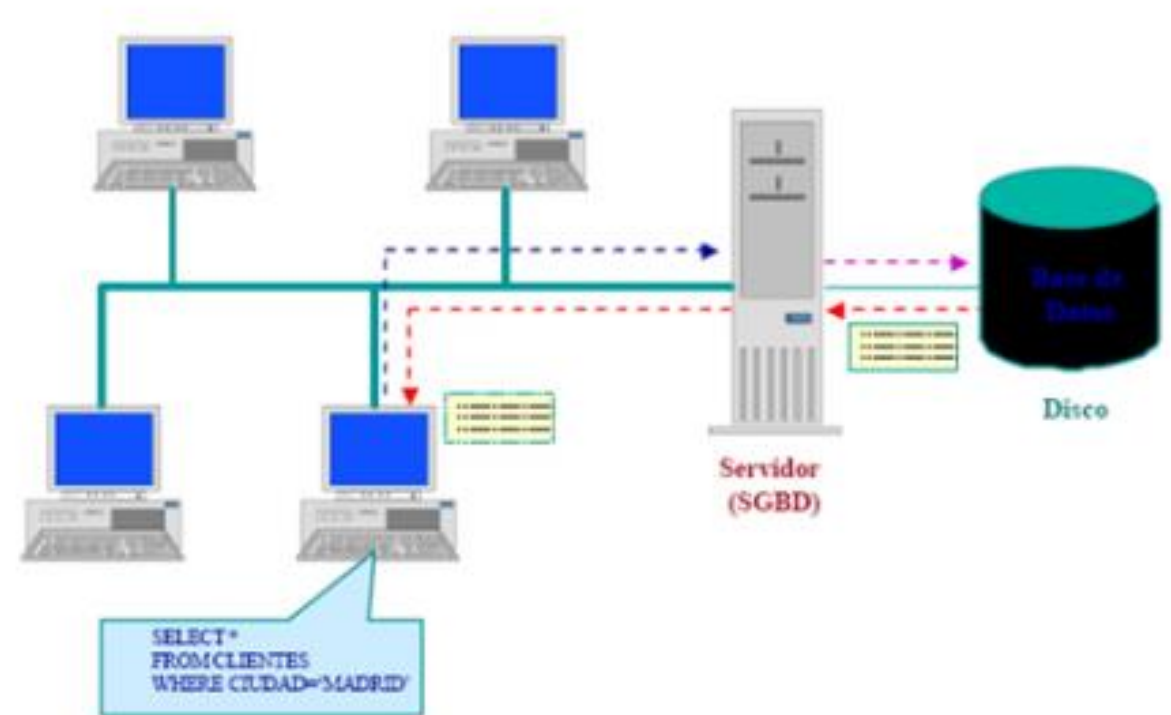
2.5 Arquitectura centralizada

- ▶ Un SABD **centralizado** combina todos sus componentes en un solo sistema, incluyendo el software del SABD, hardware, programas de aplicación, y las interfaces de usuario del procesamiento de datos.
- ▶ Un usuario se puede conectar mediante una interfaz remota, con una terminal, pero el procesamiento se hace de forma centralizada. La máquina tiene la **presentación**, la **lógica** y los **datos** del sistema de base de datos.



Arquitectura C/S

- ▶ La arquitectura **cliente servidor** C/S consta de uno (o varios) servidor(es) que proporcionan la funcionalidad principal del sistema, y de varios clientes autónomos, los cuales satisfacen sus necesidades mediante peticiones de operaciones de consultas y actualizaciones remotas al servidor.



Cientes

- ▶ Proporcionan las interfaces de software apropiadas para acceder y utilizar los recursos del servidor.
- ▶ Puede contar con recursos propios (discos, software, periféricos, etc.).
- ▶ Se conectan vía algún tipo de red de comunicaciones (LAN, Wireless, etc.).
- ▶ Pueden tener programas de software de visualización de datos (**capa de presentación**).



Servidor

- ▶ Proporcionan los servicios de consulta y actualizaciones de datos por parte de los clientes.
- ▶ Los clientes se comunican con el servidor con algún tipo de API (Application Programming Interface) estándar, como ODBC, JDBC u otra.
- ▶ Contiene la funcionalidad (**capa de lógica del negocio**) y los datos (**capa de servicios de datos**) del sistema, los cuales pueden estar soportados completamente por el SABD o divididos en dos o más capas (servidor Web y servidor de BD).



Clasificación de los SABD

- ▶ Existen varias formas de clasificación de los SABD; entre las más empleadas tenemos:
 - ▶ Por el **modelo de datos** (tradicional y emergente)
 - ▶ Por el **número de usuarios** (un solo usuario vs. multiusuarios)
 - ▶ Por la **arquitectura de comunicación** (centralizada vs. distribuida)
 - ▶ Por la **comercialización** (libres vs. comerciales)
 - ▶ Por el **costo** (gratis vs. alto)
 - ▶ Por el **tipo de procesamiento** (propósito general, OLTP (OnLine Transaction Processing), OLAP (OnLine Analytical Processing))



2.6 Breve historia de las bases de datos (1)

▶ 1960s:

- ▶ principios de los 60: Charles Bachmann desarrolla el primer SABD en Honeywell (IDS)
 - ▶ usa el modelo de red, donde las relaciones son representadas como una gráfica
- ▶ finales de 1960: primer SABD comercial exitoso desarrollado en IBM (IMS)
 - ▶ usa el modelo jerárquico donde las relaciones son representadas como un árbol
 - ▶ aún es usado actualmente (reservaciones SABRE; Travelocity)
- ▶ finales de 1960: se define el modelo CODASYL (Conference On DATA Systems Languages). Es el modelo de red, pero más estandarizado



Breve historia de las bases de datos (2)

▶ 1970s:

- ▶ 1970: Edgar F. (Ted) Codd define el modelo relacional de datos en el Laboratorio IBM de San José (ahora IBM Almaden)
- ▶ Comienzan dos grandes proyectos (que fueron operacionales al final de 1970)
 - ▶ INGRES en la Universidad de California, Berkeley
 - Se convirtió en INGRES (comercial), seguido por POSTGRES, el cuál fue incorporado en Informix
 - ▶ System R en el Laboratorio IBM de San José
 - se convirtió en DB2
- ▶ 1976: Peter Chen define el modelo Entidad-Relación (ER)



Breve historia de las bases de datos (3)

▶ 1980s

- ▶ maduración de la tecnología de bases de datos
- ▶ estandarización del SQL (mitad-final de 1980) por ISO
- ▶ periodo de real crecimiento

▶ 1990s

- ▶ continúa la expansión de la tecnología relacional y el mejoramiento del rendimiento
- ▶ la distribución se convierte en realidad
- ▶ nuevos modelos de datos: orientados a objetos, deductivos
- ▶ finales de 1990: incorporación de la orientación a objetos en los SABD relacionales
⇒ SABD objeto-relacionales
- ▶ nuevas áreas de aplicación: Dataware Housing y OLAP, Web e Internet, interés en texto y multimedia



Breve historia de las bases de datos (4)

► 2000s

- La siguiente generación de bases de datos son conocidas como noSQL, las cuales permiten almacenar valores de forma rápida de forma llave-valor, y también las bases de datos orientadas a documentos.
- Las BD NoSQL son rápidas, no requieren esquemas fijos, evitan las operaciones de reunión almacenando datos desnormalizados, y están diseñadas para escalar horizontalmente.
- Las BD XML son usadas en la administración empresarial, en donde se usa XML como mecanismo estándar de interoperabilidad entre máquinas.

