

# **6903 Bases de Datos**

---

## **El Entorno de las Bases de Datos**

Thomas M. Connolly  
Carolyn E. Begg

4ª edición

# Sistemas de bases de datos

Un enfoque práctico para diseño, implementación y gestión



# Objetivos

---

- ☐ Propósito y origen de la arquitectura en tres niveles de las bases de datos.
- ☐ El contenido de los niveles externo, conceptual e interno.
- ☐ Propósito de las asignaciones entre los niveles externo/conceptual y conceptual interno.
- ☐ El significado de la independencia de datos lógica y física.
- ☐ Distinción entre DDL y DML.
- ☐ Una clasificación de los modelos de datos.

# Objetivos

---

- ☐ Finalidad e importancia del modelado conceptual
- ☐ Los servicios y funciones típicas que un SMDB debe proporcionar.
- ☐ Función e importancia del catálogo del sistema.
- ☐ Componentes de software de un SMBD.
- ☐ Significado de la arquitectura cliente/servidor y las ventajas para un SMBD.
- ☐ La función y uso de los monitores de procesamiento de transacciones.

# Arquitectura ANSI-SPARC

---

- ❑ American National Standards Institute, Standards Planning And Requirements Committee (**ANSI-SPARC**)
- ❑ Modelo de tres niveles de la arquitectura de SMBD.
- ❑ Propuesto en 1975 para fomentar la independencia entre datos y aplicaciones.
- ❑ La mayoría de los DBMS comerciales modernos se basan en este sistema. Sin embargo, nunca se convirtió en un estándar formal.

# Objetivos de la arquitectura de tres niveles

---

- ❑ Todos los usuarios pueden tener acceso a los mismos datos.
- ❑ La vista de un usuario es inmune a los cambios en otras vistas.
- ❑ Los usuarios no necesitan conocer los detalles del almacenamiento físico de la base de datos.

# Objetivos de la arquitectura de tres niveles

---

- ❑ **Permite vistas de usuario independientes y personalizadas:** Cada usuario debe ser capaz de acceder a los datos, pero tiene una vista personalizada diferente de los datos. Estos deben ser independientes: los cambios en una vista no deben afectar a las demás.
- ❑ **Oculto los detalles físicos de almacenamiento a los usuarios:** Los usuarios no deberían tener que lidiar con los detalles de almacenamiento de la base de datos.

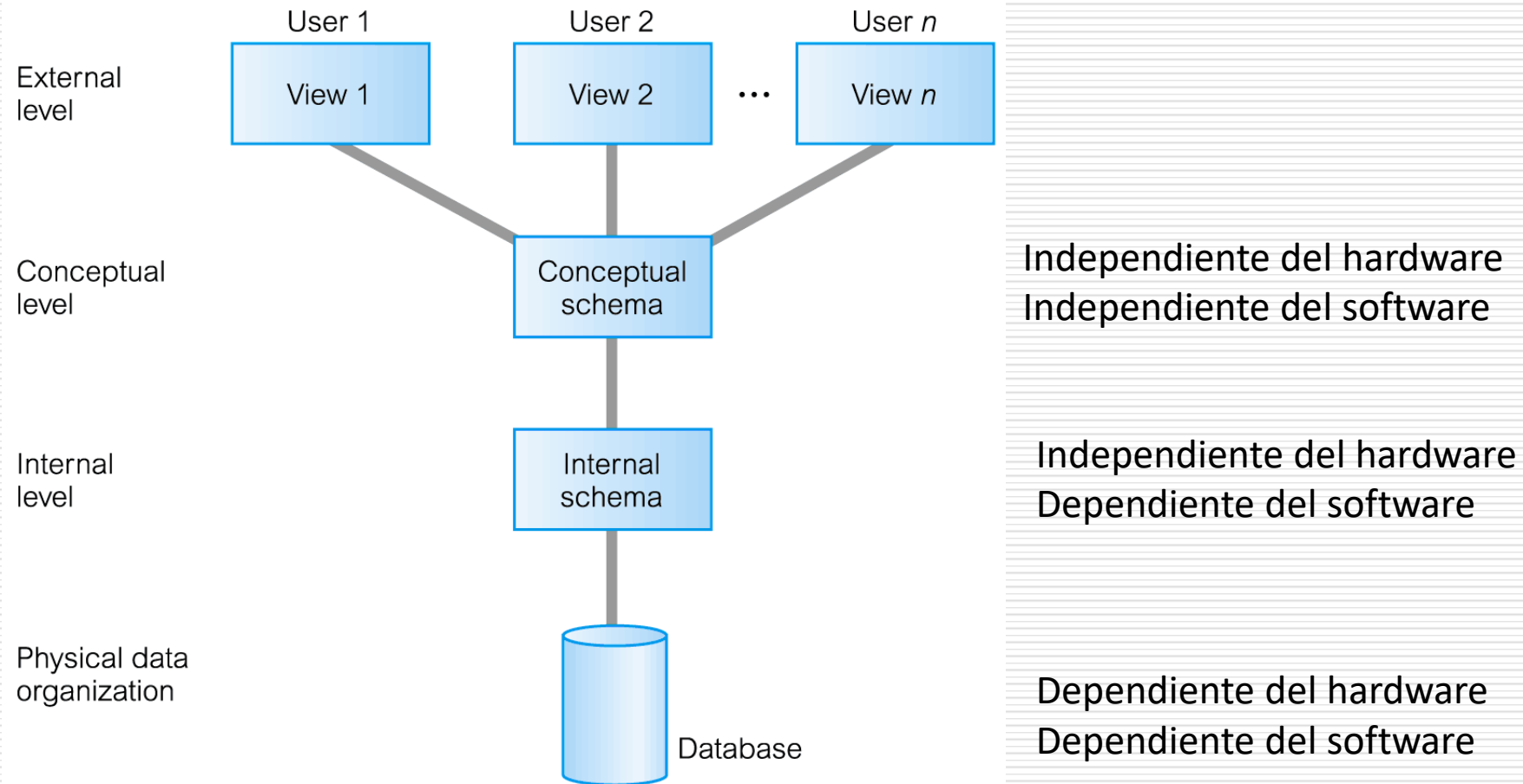
# Objetivos de la arquitectura de tres niveles

---

- ❑ El administrador de la base de datos debe ser capaz de cambiar las estructuras de almacenamiento de esta sin afectar la vista de los usuarios.
- ❑ La estructura interna de la base de datos no debería verse afectada por cambios en los aspectos físicos del almacenamiento: por ejemplo, un cambio a un nuevo disco.



# Arquitectura de tres niveles de ANSI-SPARC



# Arquitectura de tres niveles de ANSI-SPARC

---

## ☐ Nivel Externo

- Vista de los usuarios de la base de datos.
- Describe la parte de la base de datos que es relevante a un usuario en particular.

## ☐ Nivel Conceptual

- Vista comunitaria de la base de datos.
- Describe que datos se almacenan en la base de datos y las relaciones entre los datos.
  - ☐ Entidades, atributos relaciones
  - ☐ Restricciones
  - ☐ Semántica de los datos

# Arquitectura de tres niveles de ANSI-SPARC

---

## ☐ Nivel Interno

- Representación física de la base de datos en la computadora.
- Describe como están almacenados los datos en la base de datos.
  - ☐ Espacio de almacenamiento para datos e índices
  - ☐ Descripción de los registros para almacenamiento
  - ☐ Ubicación de los registros
  - ☐ Compresión y cifrado de los datos

# Diferencias entre los tres niveles de la Arquitectura ANSI-SPARC

External view 1

sNo	fName	lName	age	salary
-----	-------	-------	-----	--------

External view 2

staffNo	lName	branchNo
---------	-------	----------

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo
---------	-------	-------	-----	--------	----------

Internal level

```
struct STAFF {  
    int staffNo;  
    int branchNo;  
    char fName [15];  
    char lName [15];  
    struct date dateOfBirth;  
    float salary;  
    struct STAFF *next;           /* pointer to next Staff record */  
};  
index staffNo; index branchNo;  /* define indexes for staff */
```

# Independencia de los datos

---

## ☐ Independencia Lógica de los Datos

- Se refiere a la inmunidad de los esquemas externos a las modificaciones al esquema conceptual.
- Cambios al esquema conceptual (e.g. agregar/remover entidades).
- No debe requerir cambios al esquema externo o rescribir programas de aplicación.

# Independencia de los datos

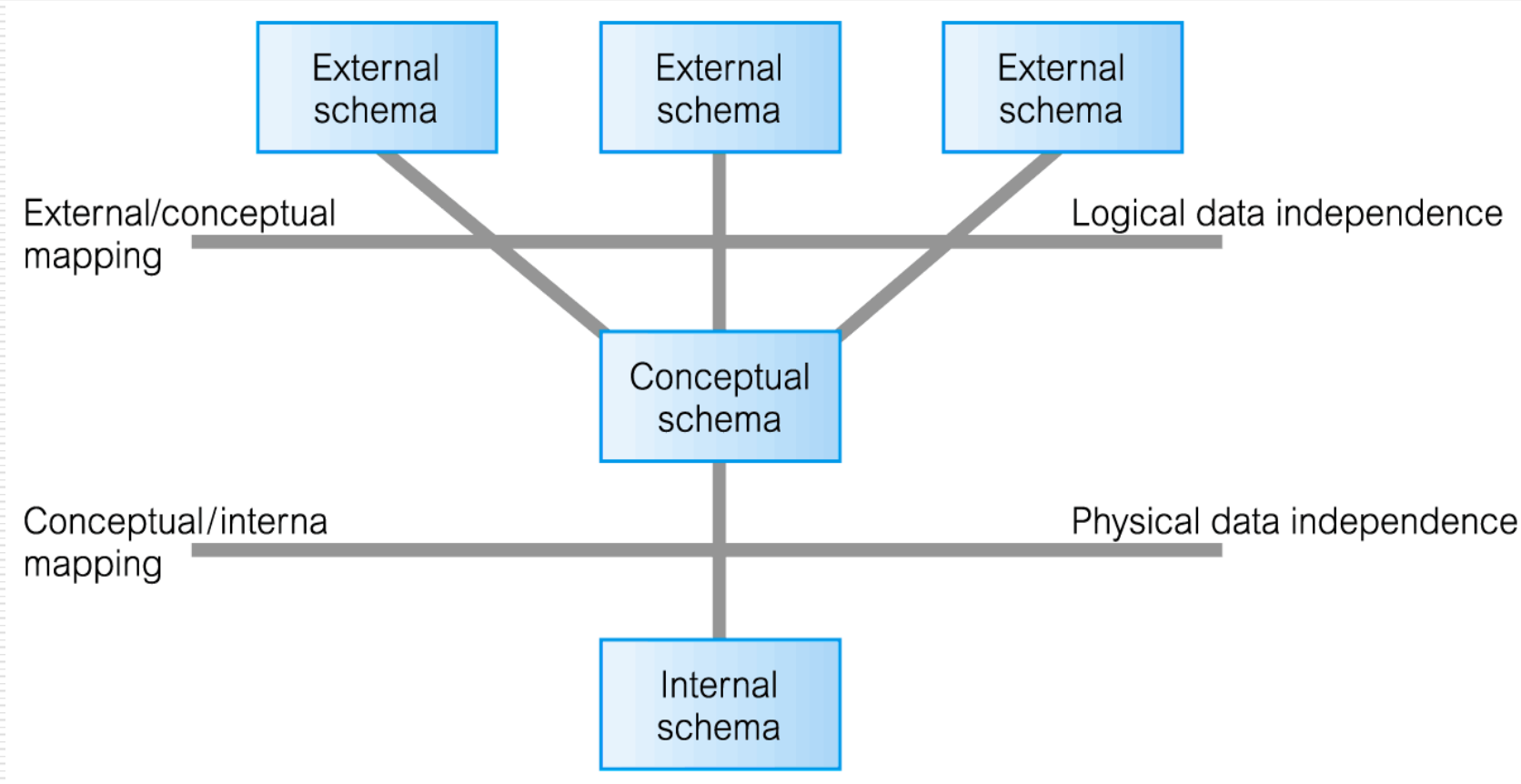
---

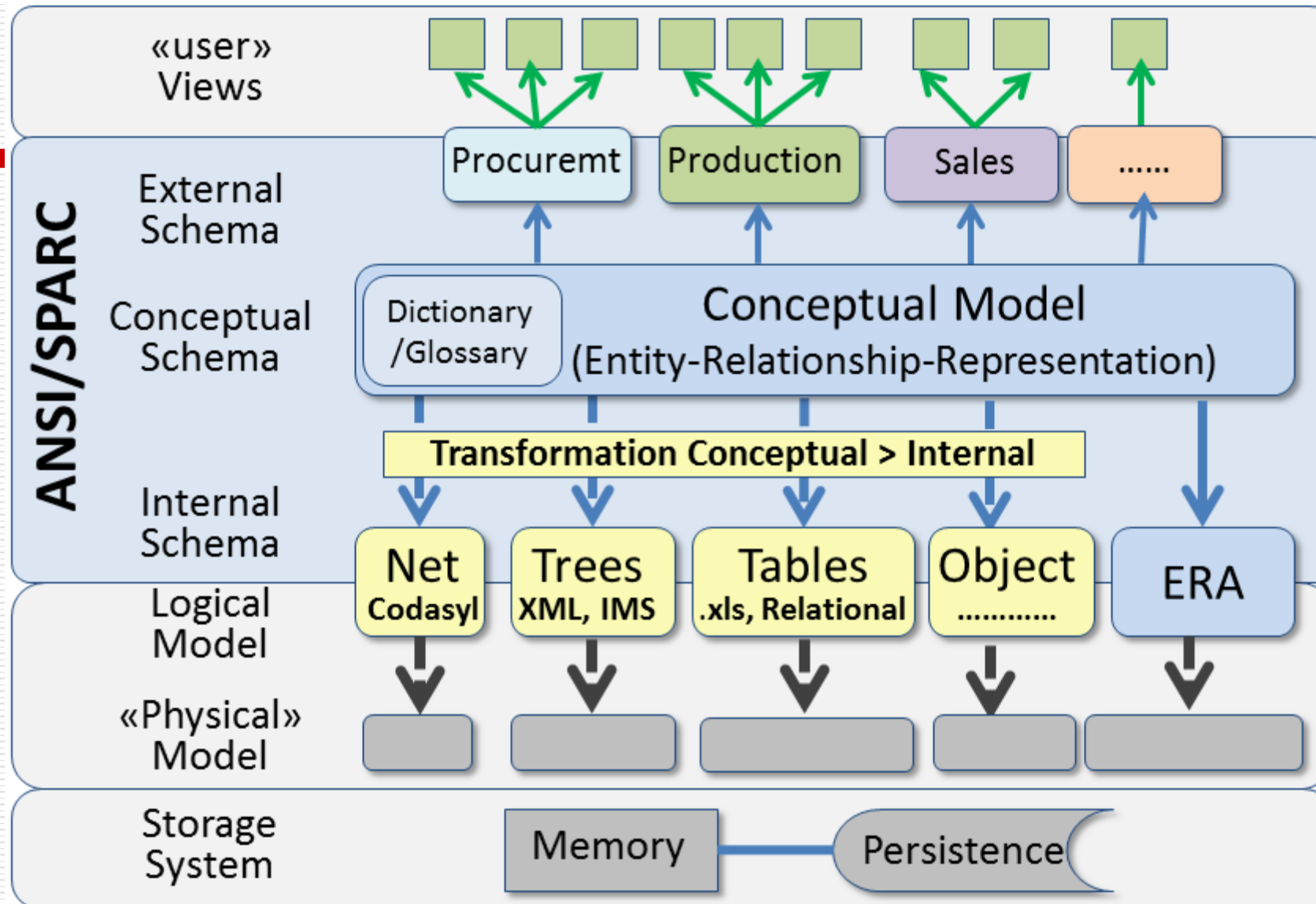
## ☐ Independencia Física de los Datos

- Se refiere a la inmunidad del esquema conceptual a los cambios en el esquema interno.
- Cambios esquema interno (e.g. Usar diferente organización de archivos, estructuras de almacenamiento o dispositivos).
- No deben requerir cambiar los esquemas conceptuales o externos.

# Independencia de los datos y la arquitectura de tres niveles de ANSI-SPARC

---







# Lenguajes de base de datos

---

## ☐ Data Definition Language (DDL)

- Permite al DBA o usuario describir y nombrar entidades, atributos y las relaciones requeridas por la aplicación.
- Adicionalmente, cualquier restricción asociada con la integridad y seguridad.

# Lenguajes de base de datos

---

## ☐ Data Manipulation Language (DML)

- Provee las operaciones de manipulación básicas de los datos contenidos en la base de datos.

## ☐ DML Procedural

- Permite al usuario indicar al sistema exactamente como manipular los datos.

## ☐ DML No-Procedural

- Permite al usuario indicar que datos se requieren en vez de como se van a recuperar.

## ☐ Fourth Generation Languages (4GLs)

# Funciones de un SMBD

---

- ❑ **Almacenamiento, recuperación y actualización de los datos.**
  - El SMBD debe proporcionar a los usuarios la capacidad para almacenar, extraer y actualizar los datos de BD.
- ❑ **Un catálogo accesible por el usuario.**
  - Descripciones de los elementos y que sea accesible por los usuarios
- ❑ **Soporte de transacciones.**
  - Debe proporcionar un mecanismo que garantice que se lleven a cabo todas las actualizaciones correspondientes a una determinada transacción, o que no se lleve a cabo ninguna.
- ❑ **Servicios de recuperación.**

# Funciones de un SMBD

---

## ☐ Servicios de control de concurrencia.

- Proporcionar un mecanismo para garantizar que la base de datos se actualice correctamente cuando haya múltiples usuarios actualizando en forma concurrente.

## ☐ Servicios de recuperación.

- Proporcionar un mecanismo para recuperar la base de datos de que ésta resulte dañada de alguna forma

## ☐ Servicios de autorización.

- Garantizar que sólo los usuarios autorizados puedan acceder a los datos.

# Funciones de un SMBD

---

## ☐ Soporte para transmitir datos.

- Comunicación con otros sistemas.

## ☐ Servicios de integridad.

- Garantizar que tanto los datos de la BD como los cambios afectados se adecuen a las reglas.

## ☐ Servicios para promover la integridad de los datos.

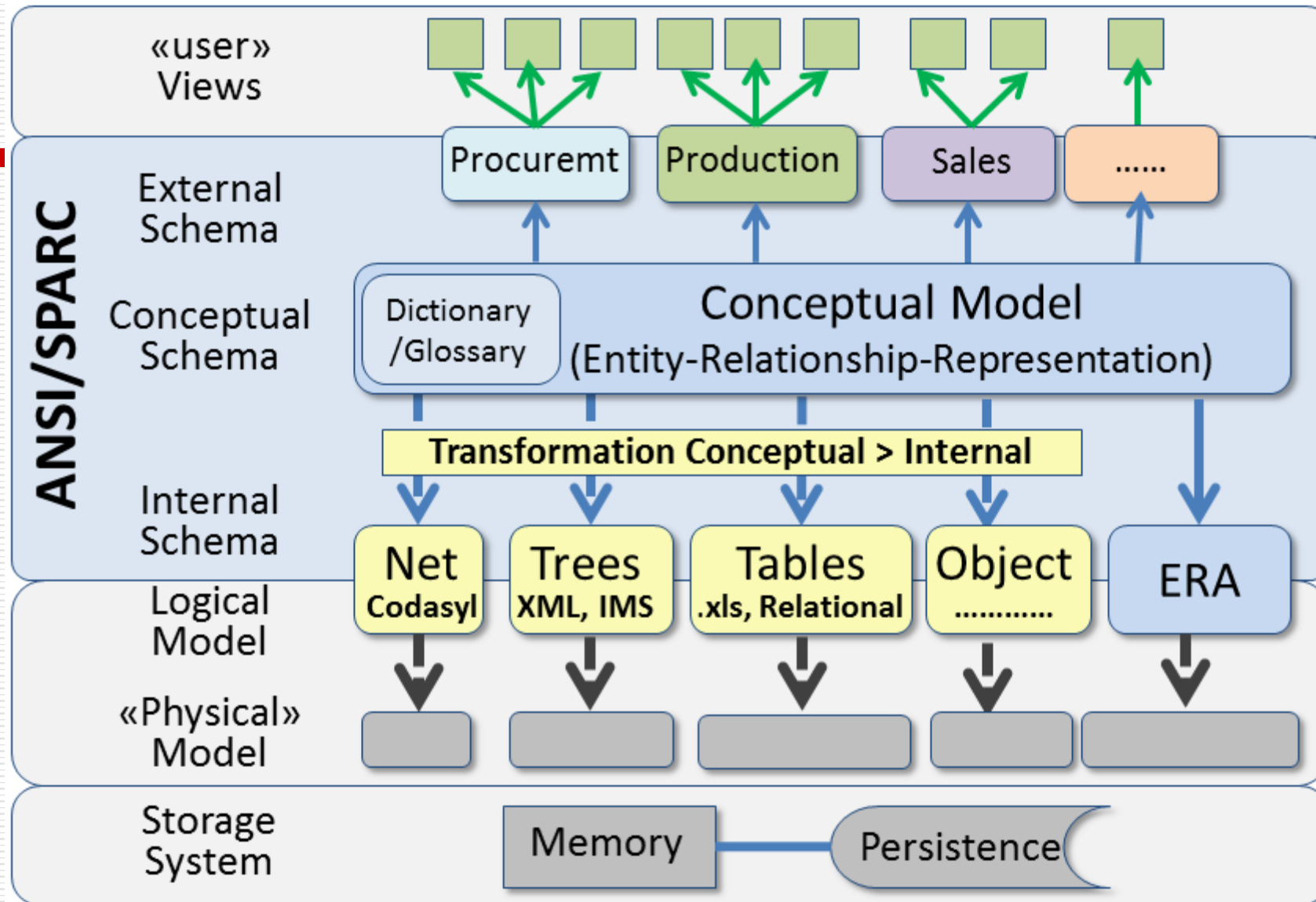
- Permitir que los programas sean independientes de la estructura real de la BD.

## ☐ Servicios de utilidad.

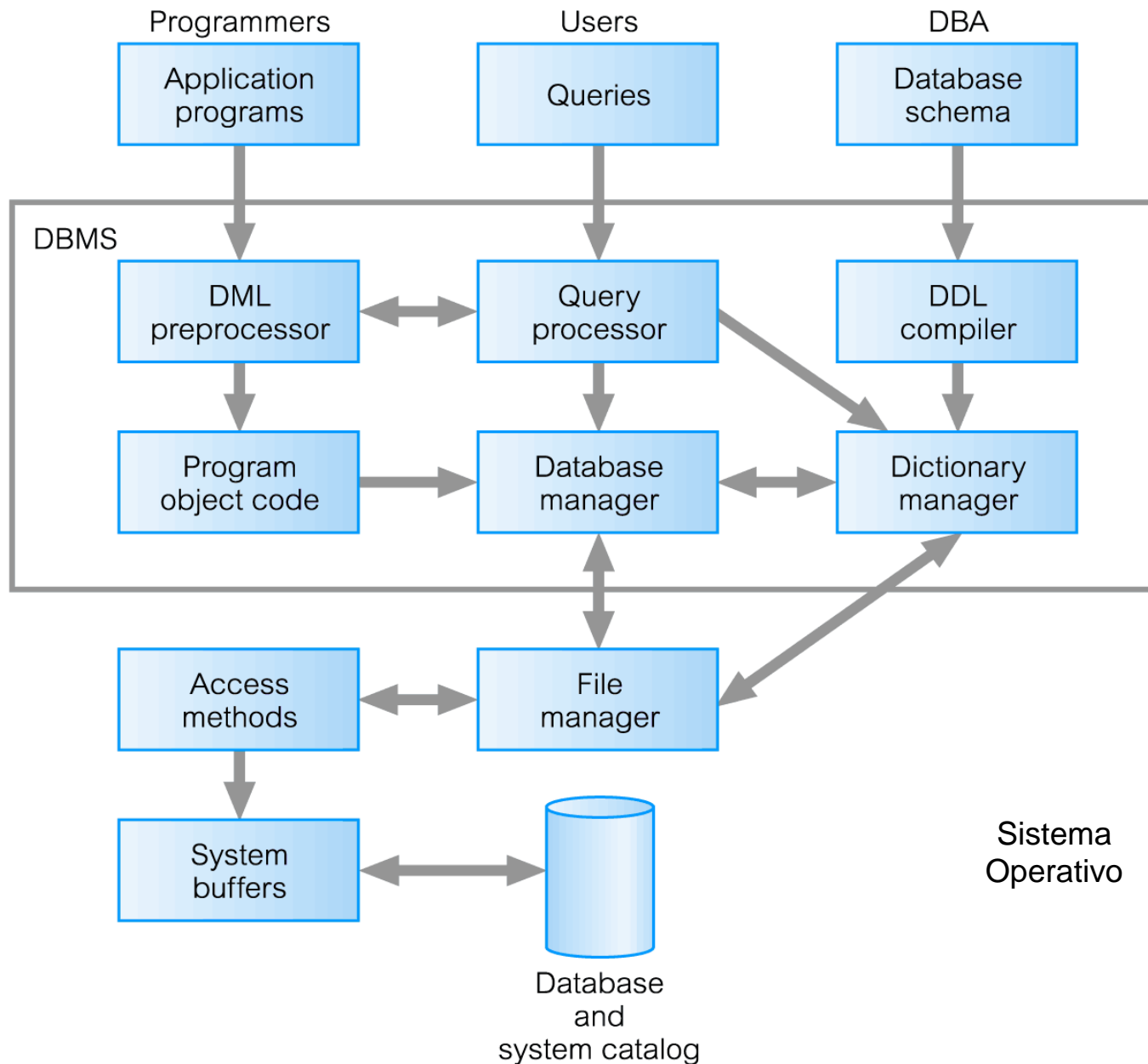
# Catálogo del sistema

---

- ❑ Es un repositorio de información que describe los datos contenidos en la base de datos (*metadatos*).
- ❑ Componente fundamental de un SMBD.
- ❑ Típicamente almacena:
  - Nombres, tipos y tamaños de los elementos de datos;
  - Restricciones en los datos;
  - Nombres de usuarios autorizados;
  - Los elementos de datos por un usuario y el tipo de acceso;
  - Estadísticas de uso.



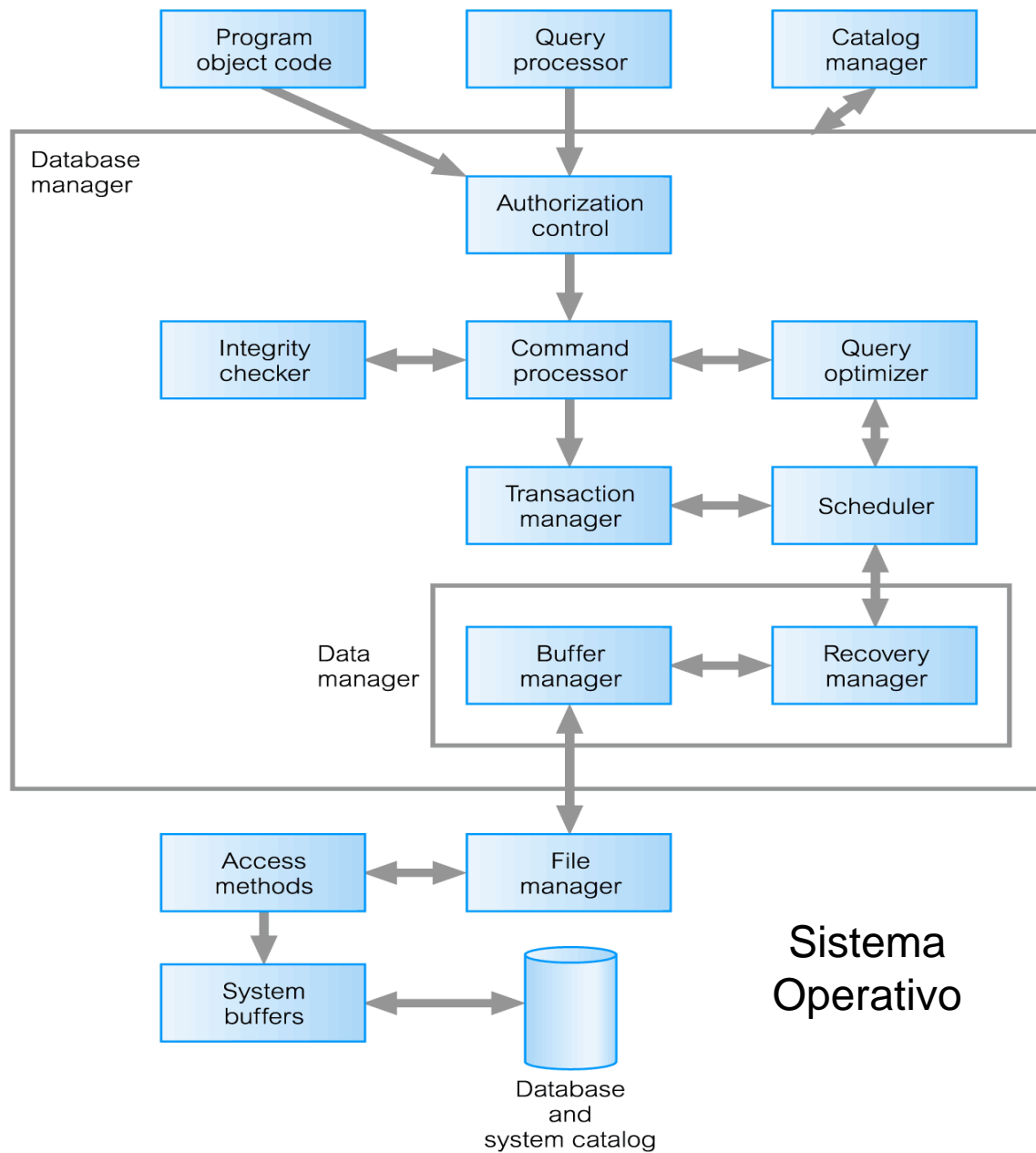
# Componentes de un SMBD



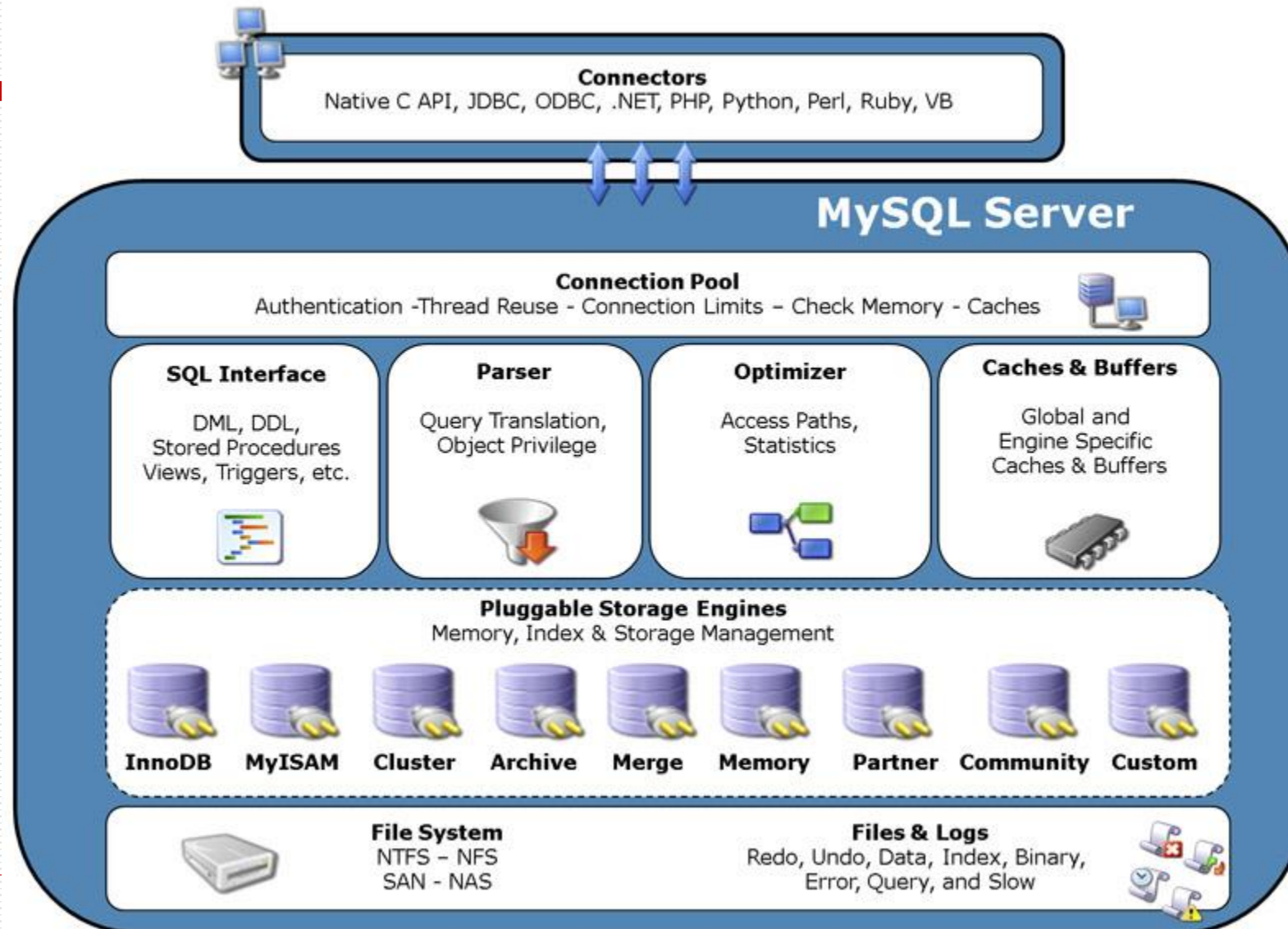
```
INSERT INTO alumnos  
VALUES (2020225, 'Juan Perez', '19990818');
```



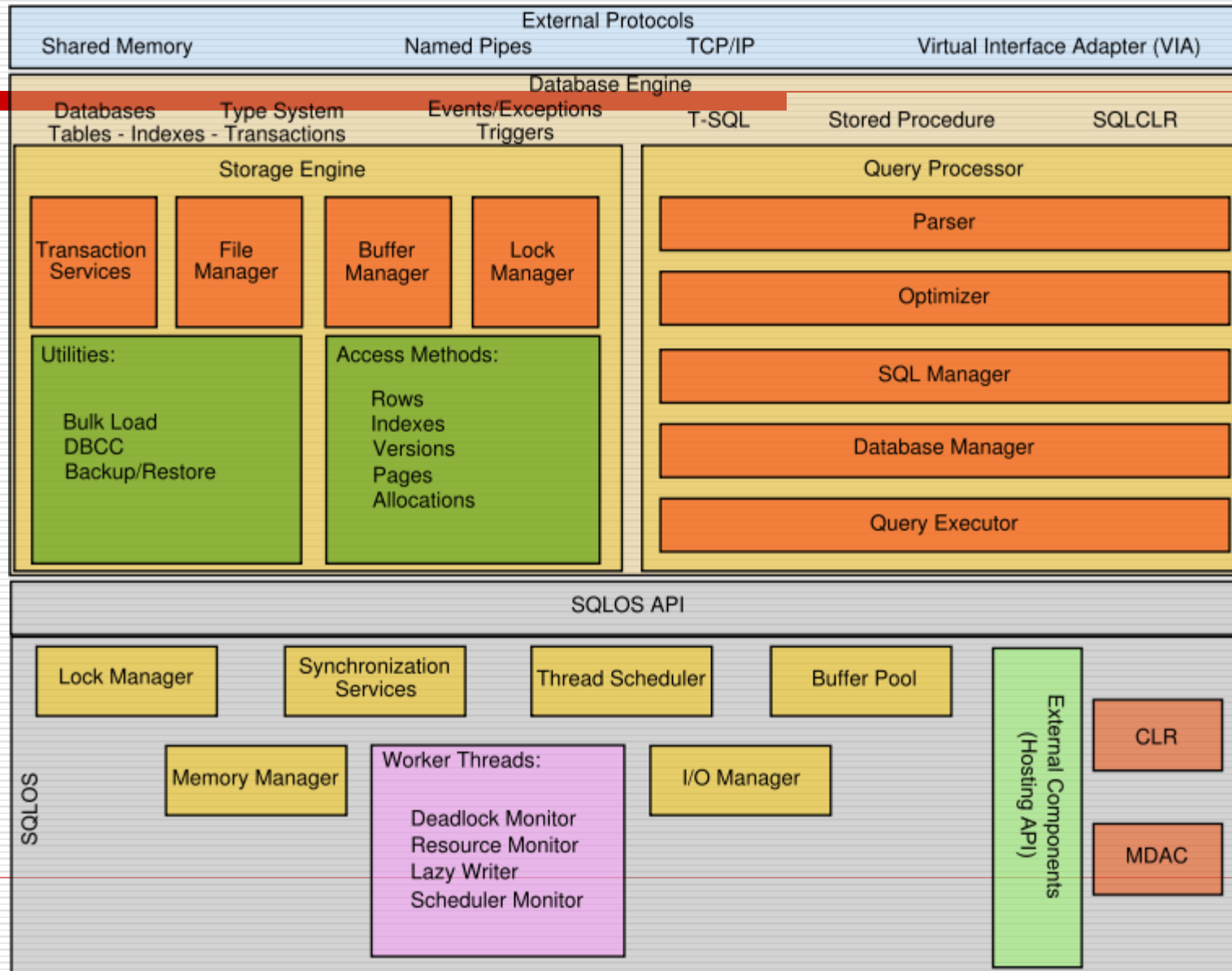
# Componentes del Administrador de Datos



# Arquitectura de MySQL



# Arquitectura de SQL Server





# Arquitecturas de SMBD multiusuarios

---

☐ Como se accesa al SMBD

☐ Teleproceso

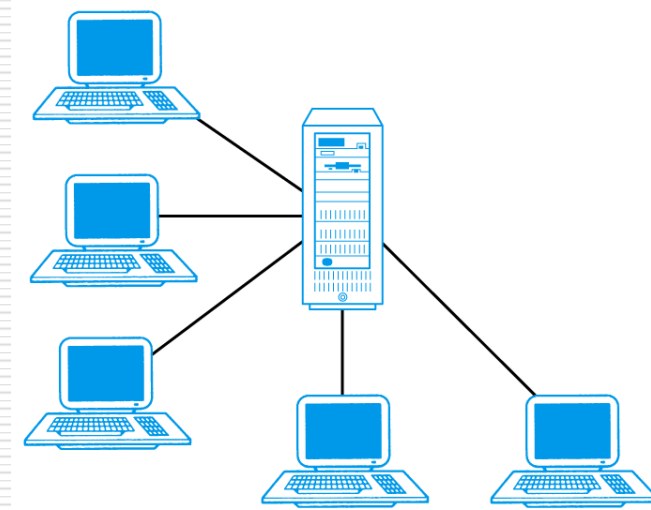
☐ Servidor de archivos

☐ Cliente/Servidor

# Teleproceso

---

- ❑ Arquitectura tradicional.
- ❑ Una mainframe con varias terminales conectadas.
- ❑ Tendencia al *downsizing*.
  - Equipo más económico.





# Servidor de archivos

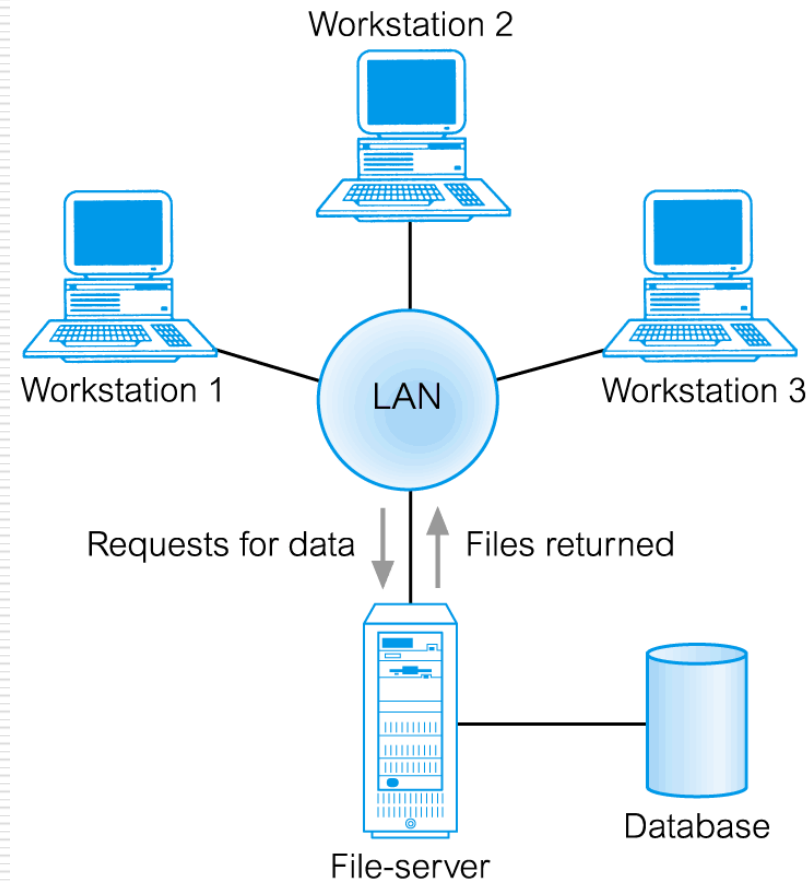
---

- ❑ El servidor de archivos se conecta a varias estaciones de trabajo a través de una red.
- ❑ La base de datos reside en el servidor de archivos.
  - El SMBD y las aplicaciones corren en cada estación de trabajo.
- ❑ Desventajas:
  - Trafico significativo en la red.
  - Copia del SMBD en cada estación de trabajo.
  - Control de la concurrencia, integridad y recuperación más compleja.



# Servidor de archivos

---



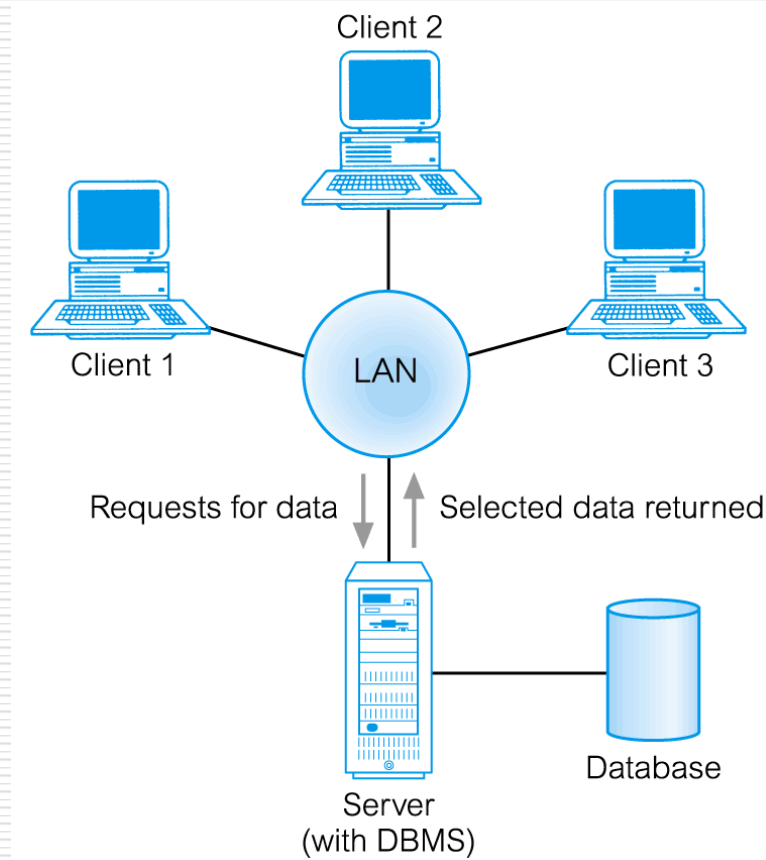
# Client-Server tradicional (Two-Tier)

---

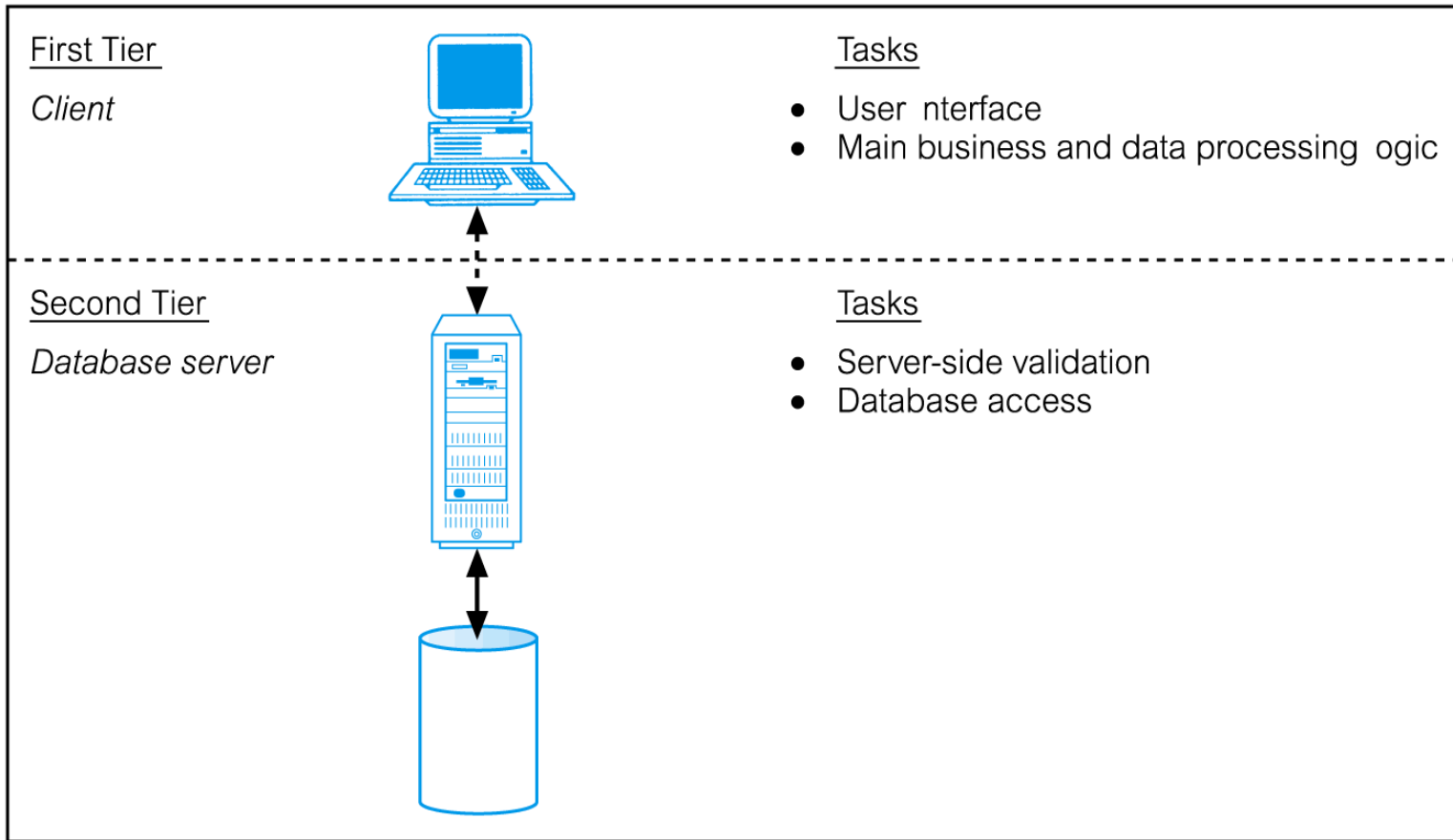
- ☐ El cliente maneja la interfaz de usuario y corre las aplicaciones.
- ☐ El servidor contiene la base de datos y el SMBD.
  
- ☐ Ventajas:
  - Acceso más amplio a las bases de datos existentes;
  - Mayor desempeño;
  - Posible reducción en costo de hardware;
  - Reducción en el costo de comunicación;
  - Se mejora la consistencia.

# Client-Server tradicional (Two-Tier)

---



# Client-Server tradicional (Two-Tier)



# Three-Tier Client-Server

---

- El lado del cliente presenta dos problemas que evitan la escalabilidad:
  - *'Fat' client*, requerimientos considerables de recursos en la máquina cliente para funcionar eficientemente.
  - Burocracia significativa para la administración de los clientes.
  
- Para 1995, se proponen tres capas, cada una corriendo potencialmente en una plataforma diferente.

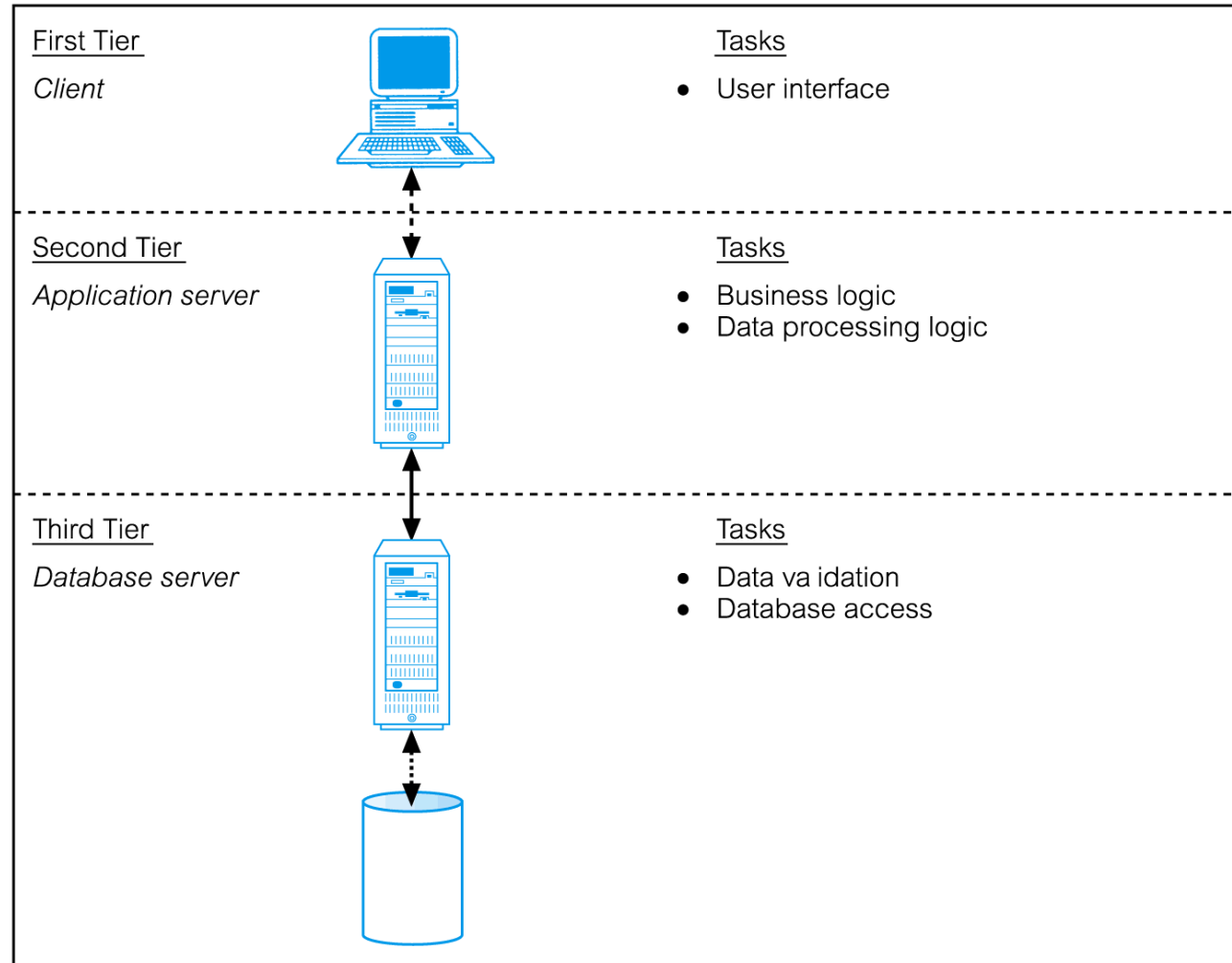
# Three-Tier Client-Server

---

## □ Ventajas:

- *'Thin' client*, que requieren hardware menos caro.
- La aplicación del mantenimiento se centraliza.
- Es más fácil modificar o remplazar una capa sin afectar a otras.
- Al separa la lógica del negocio de las funciones de base de datos hace más fácil el balanceo de carga.
- Se adapta mejor al ambiente del web

# Three-Tier Client-Server



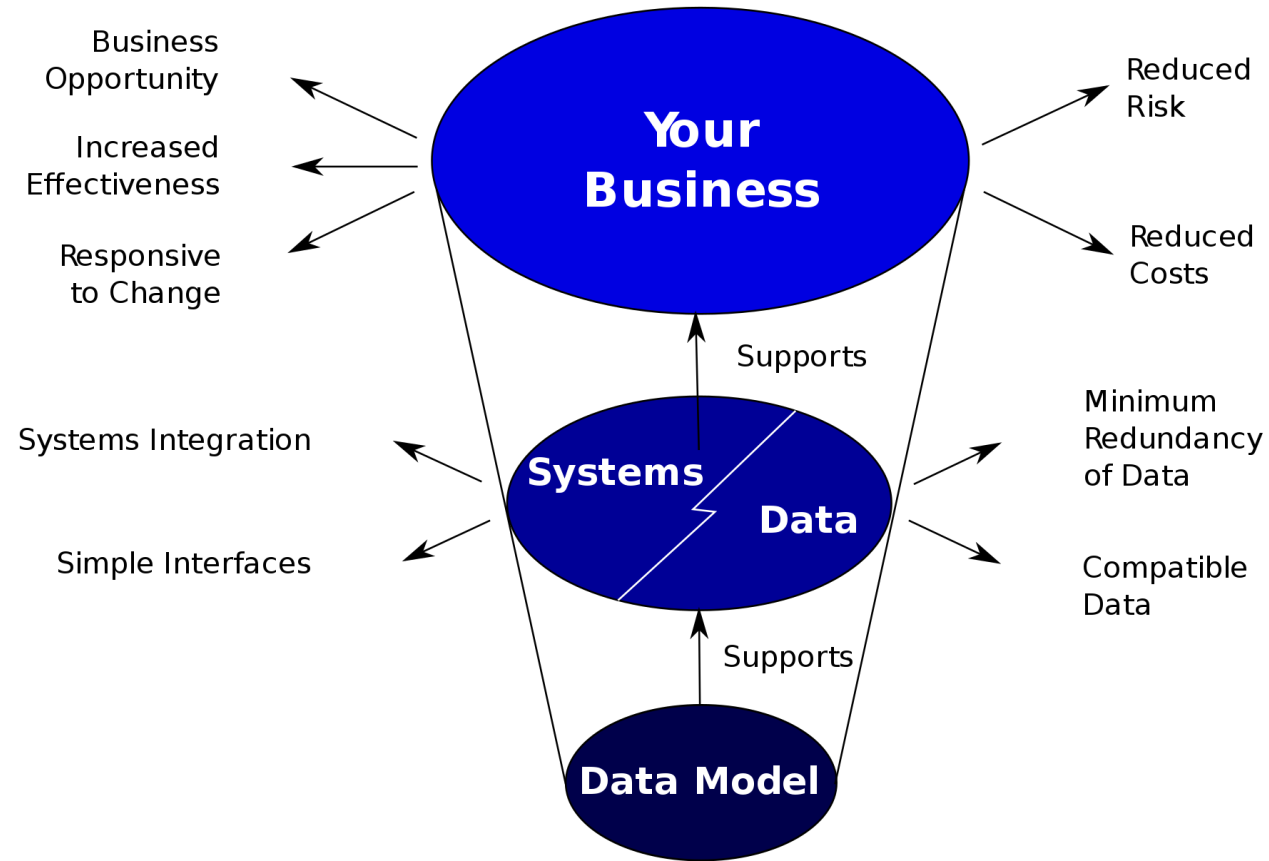
---

# MODELOS DE DATOS

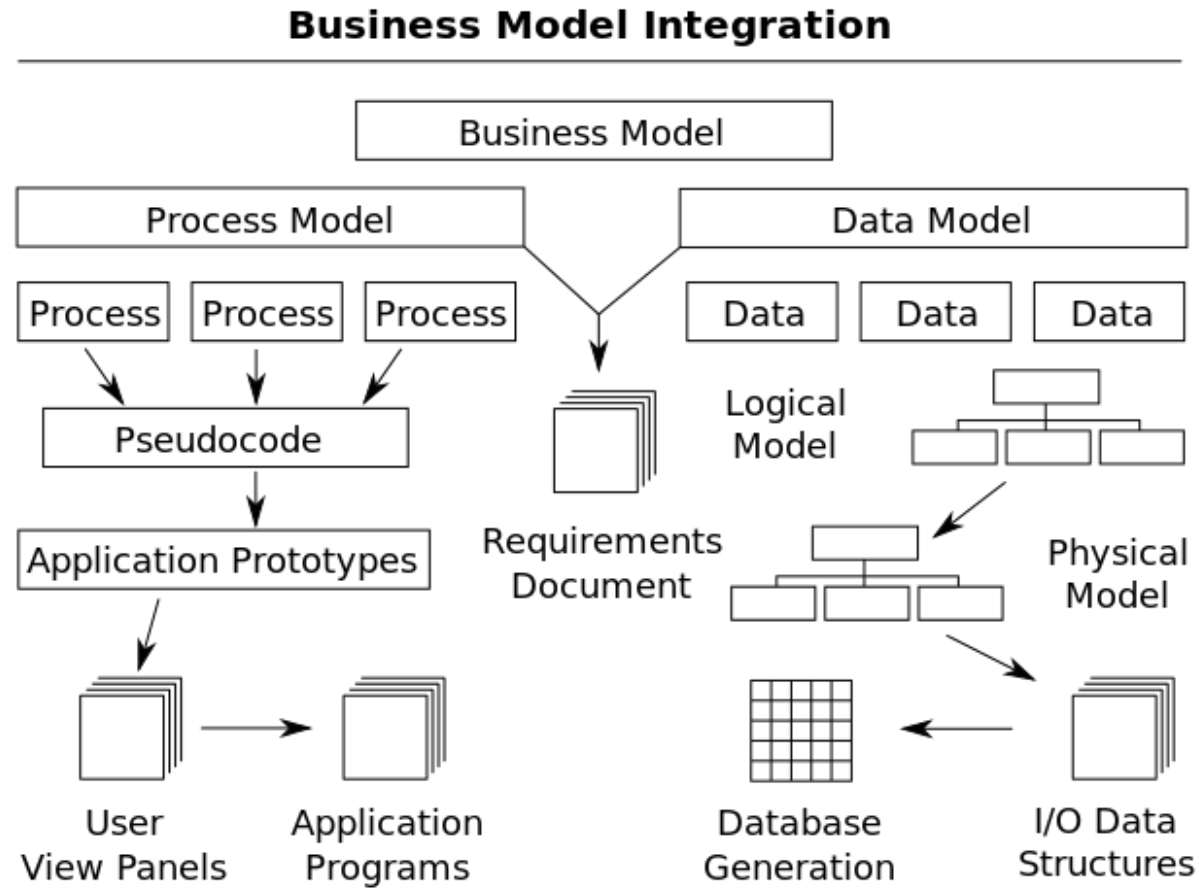


# Los procesos de negocio y los datos

---



# Los procesos de negocio y los datos



# Modelos de datos

---

- ❑ Modelo: Una abstracción de un objeto o evento del mundo real.
- ❑ Útil para entender las complejidades del mundo real.

# Modelos de datos

---

- Colección integrada de conceptos para describir y manipular datos, las relaciones existentes entre los mismos y las restricciones aplicables a los datos, todo ello dentro de una organización.
  
- Un Modelo de Datos comprende:
  - Una parte estructural;
  - Una parte operativa;
  - Posiblemente un conjunto de restricciones de integridad.

# Modelos de datos: Importancia

---

- ☐ Facilitar la interacción entre el diseñador, el programador de aplicaciones, y el usuario final.
- ☐ Los usuarios finales tienen diferentes puntos de vista y necesidades de los datos.
- ☐ Organiza los datos para varios usuarios.

# Modelos de datos

---

## ☐ Finalidad

- Representar datos en una manera entendible.

## ☐ Las categorías de modelos de datos incluyen:

- Basados en objetos
- Basados en registros
- Físicos.

# Modelos de datos

---

## ☐ Modelos de datos basados en objetos

- Entidad-Relación
- Semántico
- Funcional
- Orientado a objetos

## ☐ Modelos de datos basados en registros

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

## ☐ Modelos de datos físicos

# Evolución de los modelos de datos

TABLE  
2.1

Evolution of Major Data Models

GENERATION	TIME	DATA MODEL	EXAMPLES	COMMENTS
First	1960s–1970s	File system	VMS/VSAM	Used mainly on IBM mainframe systems Managed records, not relationships
Second	1970s	Hierarchical and network	IMS, ADABAS, IDS-II	Early database systems Navigational access
Third	Mid-1970s	Relational	DB2 Oracle MS SQL Server MySQL	Conceptual simplicity Entity relationship (ER) modeling and support for relational data modeling
Fourth	Mid-1980s	Object-oriented Object/ relational (O/R)	Versant Objectivity/DB DB2 UDB Oracle 11g	Object/relational supports object data types Star Schema support for data warehousing Web databases become common
Fifth	Mid-1990s	XML Hybrid DBMS	dbXML Tamino DB2 UDB Oracle 11g MS SQL Server	Unstructured data support O/R model supports XML documents Hybrid DBMS adds object front end to relational databases Support large databases (terabyte size)
Emerging Models: NoSQL	Late 2000s to present	Key-value store Column store	SimpleDB (Amazon) BigTable (Google) Cassandra (Apache)	Distributed, highly scalable High performance, fault tolerant Very large storage (petabytes) Suited for sparse data Proprietary API



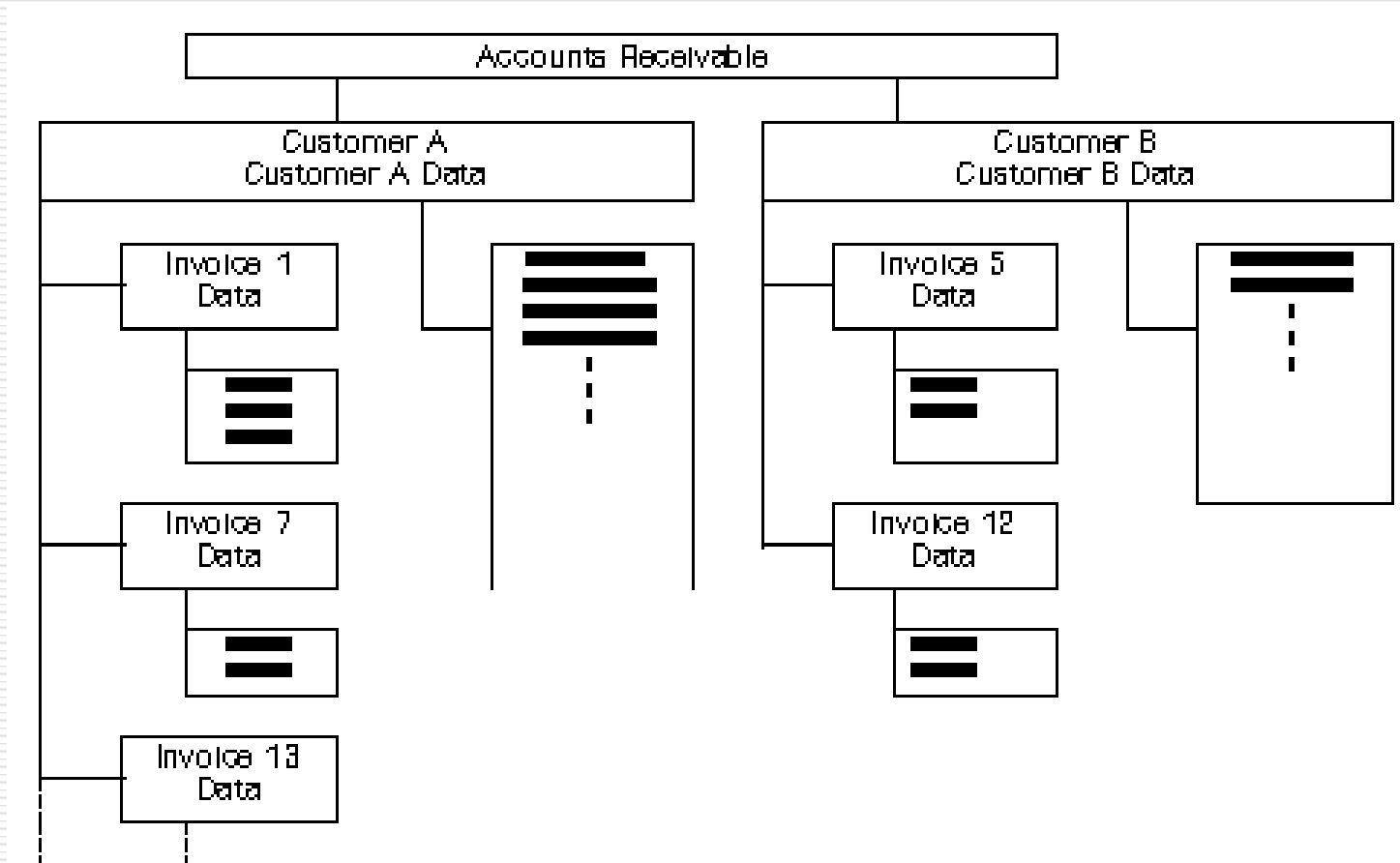
# Modelo de datos jerárquico

---

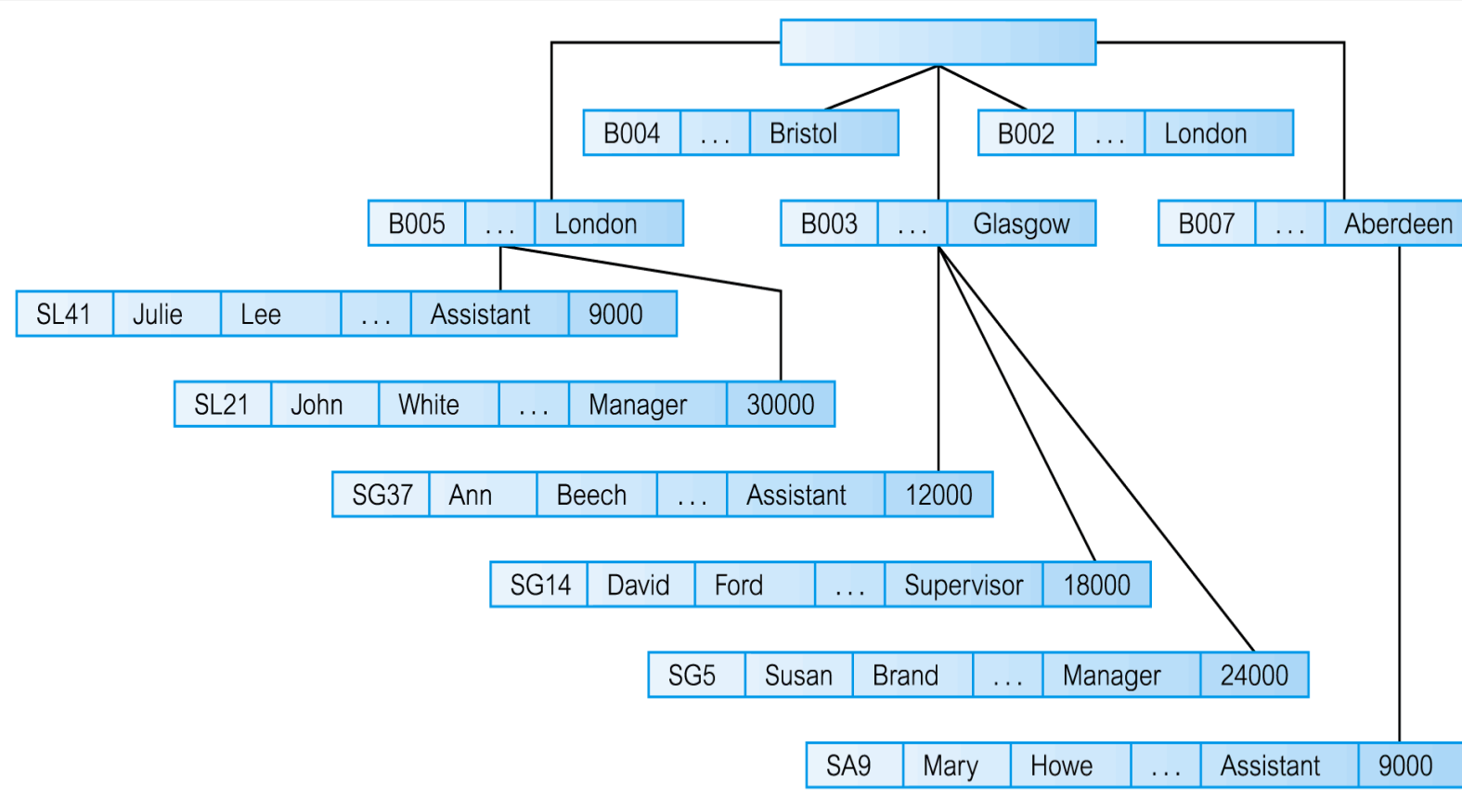
- ❑ Desarrollado en los 60s para manejar grandes cantidades de datos de proyectos de manufactura.
- ❑ La estructura lógica básica se representa por un árbol invertido.
- ❑ La estructura contiene niveles o segmentos.
- ❑ La información se presenta usando relaciones padre/hijo.
  - Cada padre tiene muchos hijos, pero cada hijo solo tiene un único padre (relaciones 1 a muchos).

# Modelo de datos jerárquico

---



# Modelo de datos jerárquico



# Modelo de datos en red

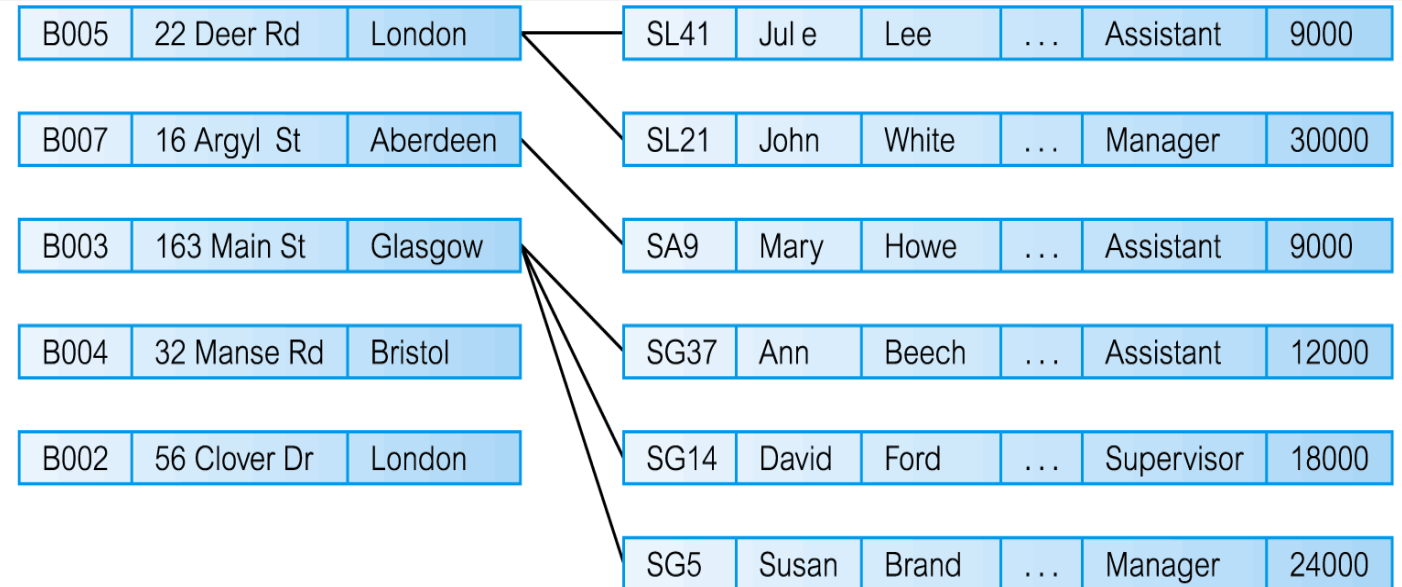
---

- ☐ Se creo para representar relaciones complejas entre los datos más eficientemente.
- ☐ Mejora el desempeño de la base de datos.
- ☐ Impone una base de datos estándar.
- ☐ Se parece al modelo jerárquico
  - El registro puede tener más de un padre

# Modelo de datos en red

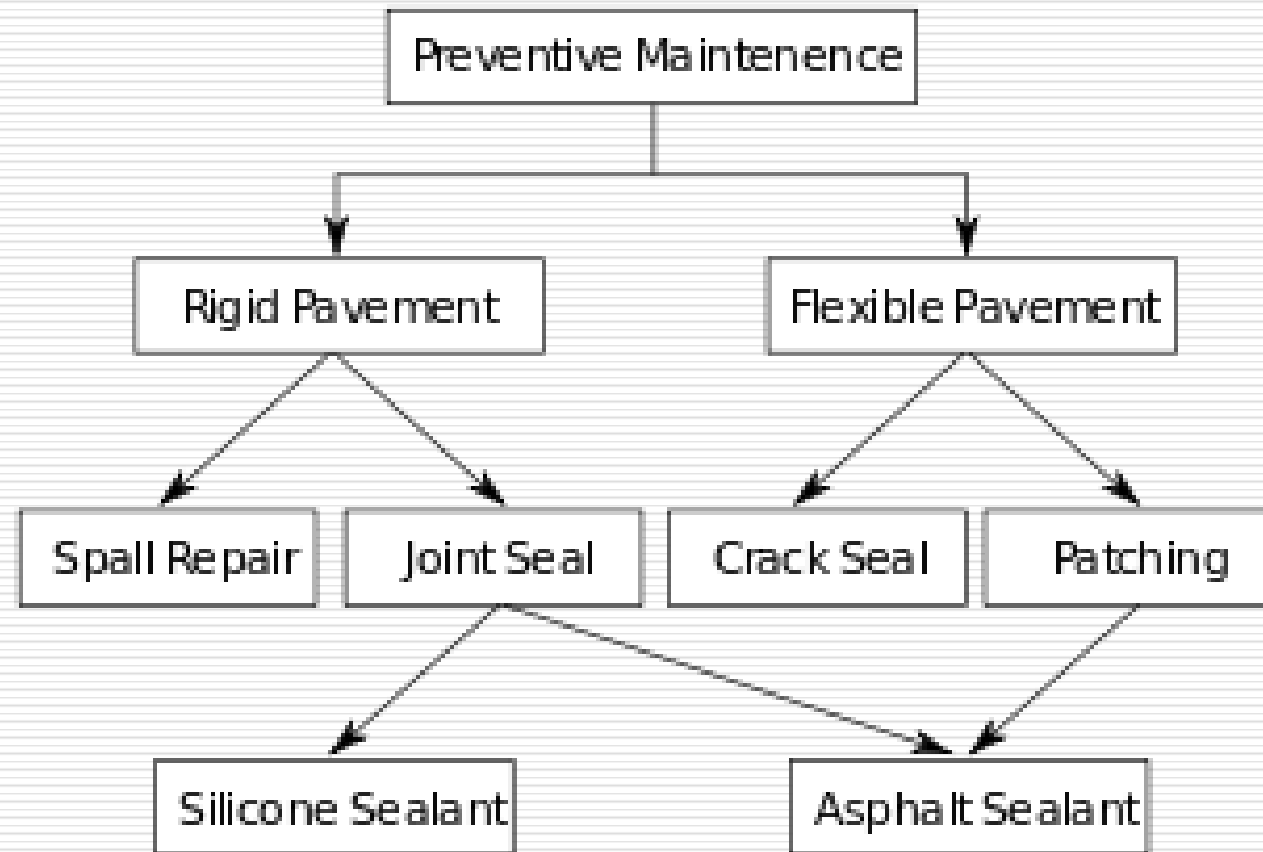
Los datos se representan como registros y las relaciones son conjuntos.

En enlace es explícito mediante punteros que forman grafos.



# Modelo de datos en red

---



# Modelo de datos relacional

---

- ❑ Desarrollado E.F. Codd (IBM) en 1970
- ❑ Tablas (Relaciones)
  - Matrices formadas por intersecciones de renglones y columnas
  - Cada renglón en una relación se le llama *tupla*.
- ❑ Se consideraban imprácticos en 1970.
- ❑ El modelo conceptualmente es simple, pero requiere más recursos computacionales.

# Modelo de datos relacional

Tablas de columnas y renglones

Branch

branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005



# Modelo de datos relacional

**FIGURE  
2.1**

**Linking relational tables**

Table name: AGENT (first six attributes)

Database name: Ch02\_InsureCo

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT\_CODE

Table name: CUSTOMER

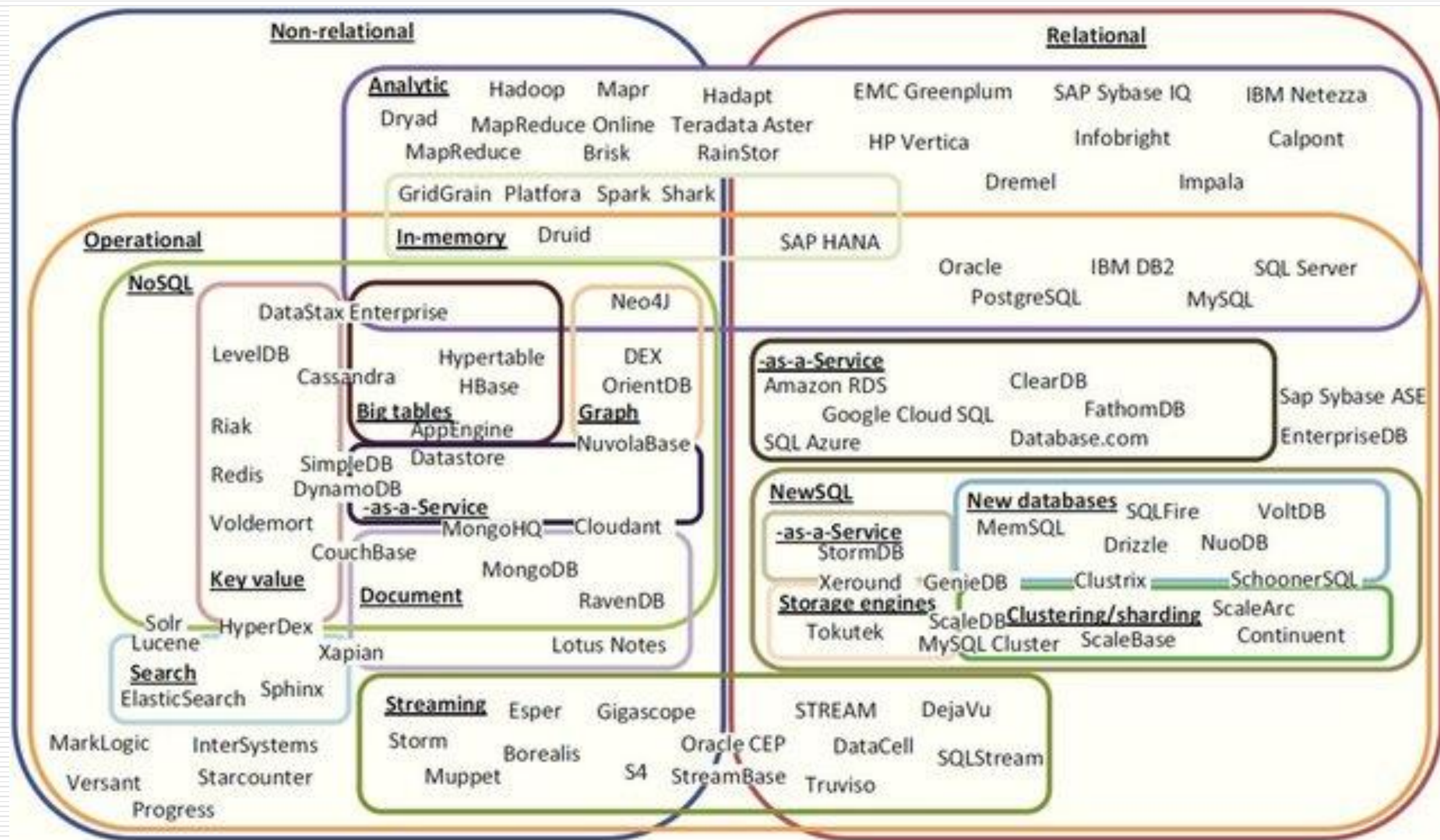
CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_INSURE_TYPE	CUS_INSURE_AMT	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	T1	100.00	05-Apr-2012	502
10011	Dunne	Leona	K	713	894-1238	T1	250.00	16-Jun-2012	501
10012	Smith	Kathy	W	615	894-2285	S2	150.00	29-Jan-2013	502
10013	Olowski	Paul	F	615	894-2180	S1	300.00	14-Oct-2012	502
10014	Orlando	Myron		615	222-1672	T1	100.00	28-Dec-2013	501
10015	O'Brian	Amy	B	713	442-3381	T2	850.00	22-Sep-2012	503
10016	Brown	James	G	615	297-1228	S1	120.00	25-Mar-2013	502
10017	Williams	George		615	290-2556	S1	250.00	17-Jul-2012	503
10018	Farriss	Anne	G	713	382-7185	T2	100.00	03-Dec-2012	501
10019	Smith	Olette	K	615	297-3809	S2	500.00	14-Mar-2013	503

SOURCE: Course Technology/Cengage Learning

# Otros modelos de bases de datos

---

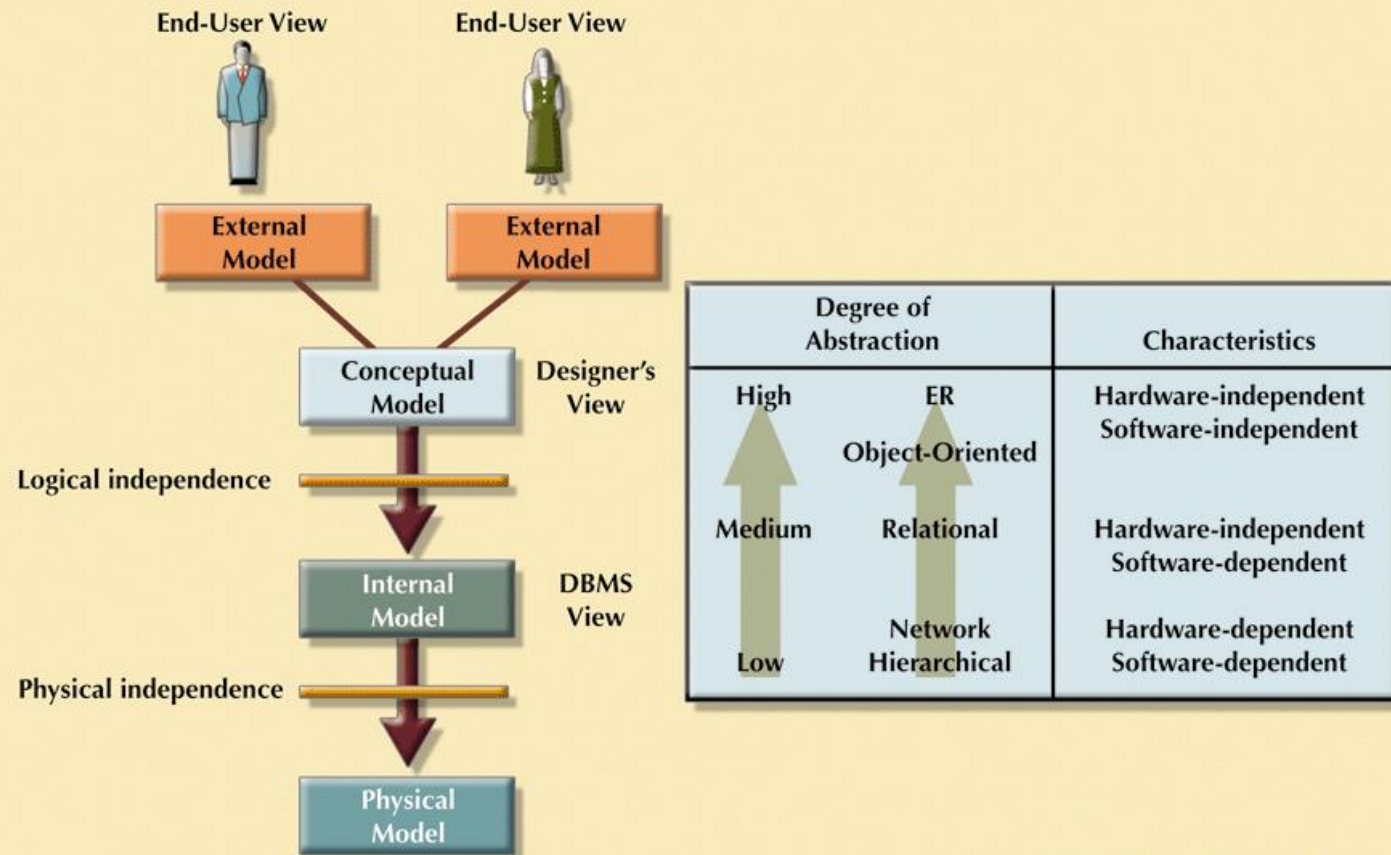
- ☐ Modelo orientado a objetos
- ☐ Semántico/XML
- ☐ Modelo noSQL/Big data



# Niveles de abstracción en el modelo ANSI-SPARC

FIGURE 2.7

Data abstraction levels



SOURCE: Course Technology/Cengage Learning

# Modelo Externo

---

- Vista de los datos del ambiente de los usuarios finales
- Representación gráfica/visual
  - E.G. Diagramas UML, ERD
- **Esquema externo:** Representación específica de la vista externa
  - Entidades
  - Relaciones
  - Procesos
  - Restricciones

# Modelo Externo

---

- ❑ Útil para identificar datos específicos que se requieren para las operaciones de cada unidad de un negocio
- ❑ Facilita el trabajo del diseñador proporcionando retroalimentación acerca del modelo
- ❑ Describe restricciones de seguridad en el diseño de la base de datos
- ❑ Simplifica el desarrollo de programas de aplicación



# Modelado conceptual

---

- ❑ El esquema conceptual es el corazón del sistema que apoya todas las vistas de los usuarios.
- ❑ Debe ser una representación completa y precisa de las necesidades de información de la empresa.
- ❑ El modelado conceptual es el proceso de construir un modelo del uso de la información en la empresa que sea independiente de los detalles de implementación.
- ❑ El resultado es el *modelo de datos conceptual*.

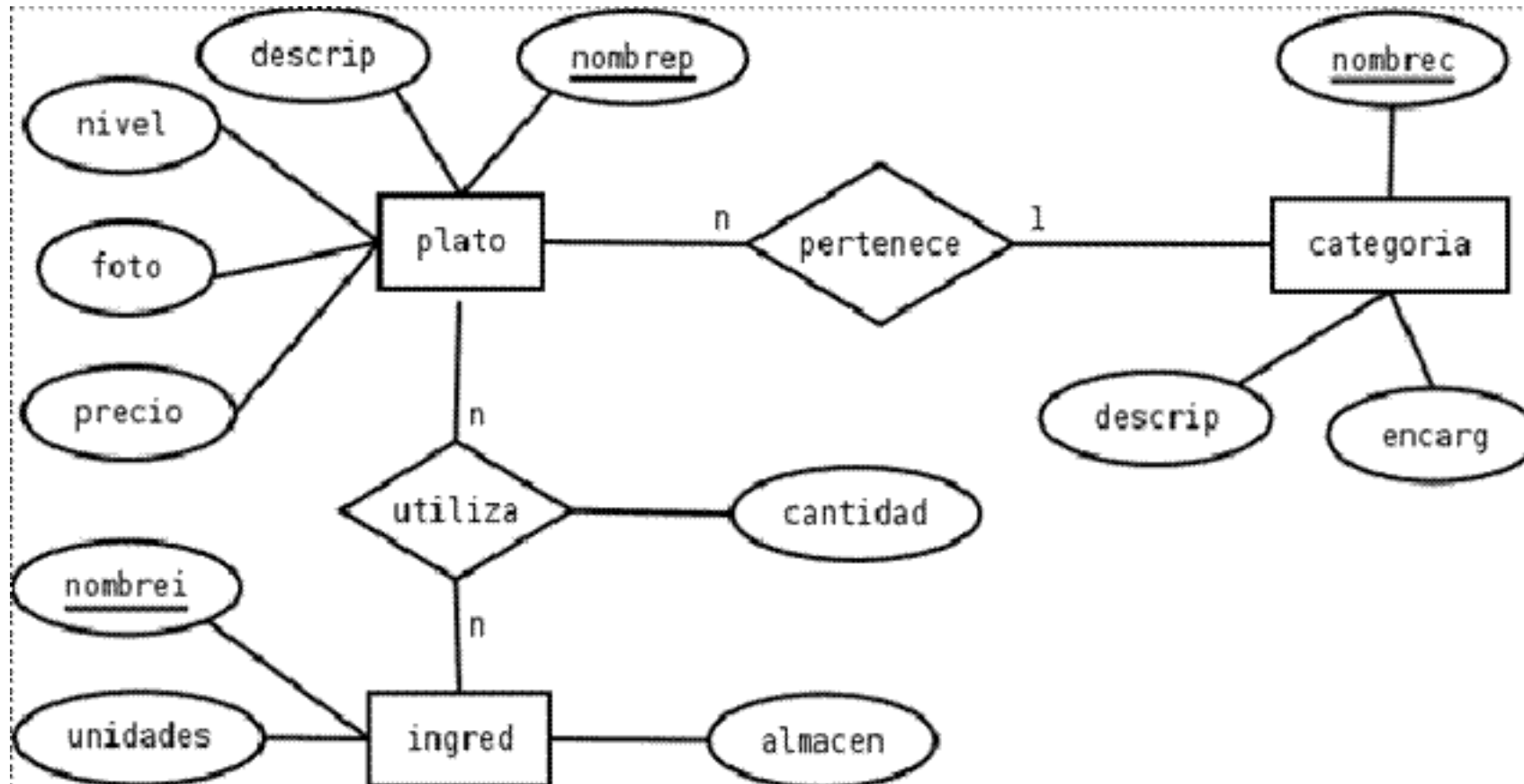
# Modelado conceptual

---

- Provee una vista a nivel macro relativamente fácil de entender del ambiente de datos
- Independiente de hardware y software
  - No depende del SMBD usado para implementar el modelo
  - No depende del hardware usado para implementar el modelo
  - Cambios en hardware/software no afectan el diseño de la base de datos a nivel conceptual



# ¿Qué representa el diagrama?



# Modelo interno

---

- Representación de la base de datos como la ve el SMBD
  - Mapea el modelo conceptual al SMBD
- El esquema interno describe una representación específica de un modelo interno.
- Depende de un software de bases de datos específico
- *Independencia lógica*: cambiar el modelo interno sin afectar el modelo conceptual.

# Modelo físico

---

- ❑ Opera al nivel más bajo de abstracción
  - Describe como se almacenan los datos en dispositivos de almacenamiento
- ❑ Requiere la definición métodos de almacenamiento físico y de acceso a los datos
- ❑ Modelo relacional se enfoca al nivel lógico
- ❑ Independencia física: Cambios en el modelo físico no afectan el modelo interno