



# **Introducción a las Bases de Datos TUPAR**

---

**Cursada 2008**

## **Clase 1: Introducción**

Facultad de Ciencias Exactas  
Universidad Nac. Centro de la Pcia. de Bs. As.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***INTRODUCCION***

---

### ***Sistemas de archivos.***

Creados sin seguir un enfoque metodológico producen:

Duplicación de datos → datos redundantes.

Separación y aislamiento de datos.

Distintos formatos para los mismos datos.

Dependencia de datos → Programas dependientes de la estructura física de los archivos.

Formatos de archivos incompatibles.

Consultas fijas / proliferación de programas de aplicación.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***BASE DE DATOS:***

---

***Base de datos:*** conjunto de datos almacenados inter-relacionados.

Diseñada para satisfacer los requerimientos de información de una organización.

Almacena:

- los datos de la organización y
- una descripción de dichos datos (datos acerca de los datos)

***metadatos → diccionario de datos***

**independencia de datos lógica – física.**



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***BASE DE DATOS:***

---

Sistemas de bases de datos → Surgen a causa de los problemas que plantean los sistemas de archivos.

- Punto de vista lógico (usuarios):
  - almacenan todos los datos en un solo lugar
  - almacenan definiciones de las relaciones entre los datos.
- Punto de vista físico:
  - los datos se almacenan en uno o varios discos y en uno o varios archivos (aún en una o varias locaciones).
- Acceso a datos:
  - a través del DBMS (Database Management System) y mediante sentencias específicas (lenguajes de cuarta generación).

## **ARQUITECTURA EN TRES NIVELES**

# INTRODUCCIÓN A LAS BASES DE DATOS

## ARQUITECTURA EN TRES NIVELES

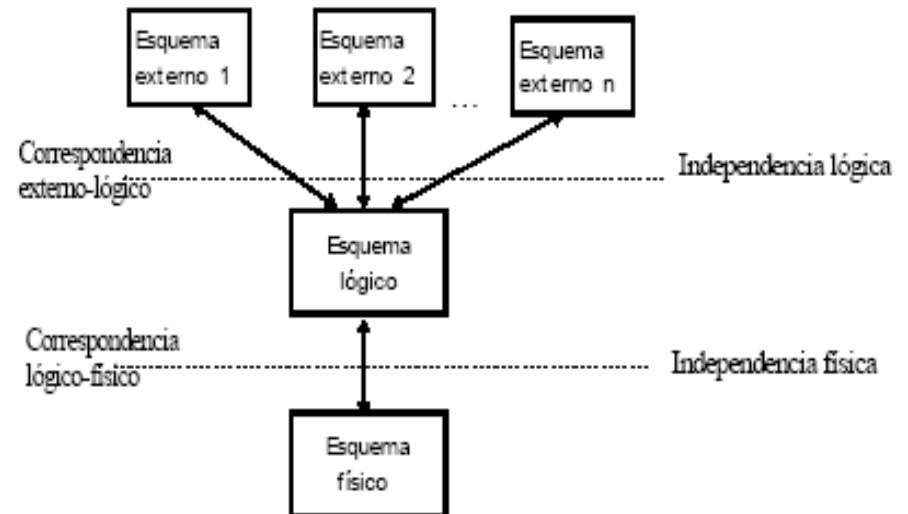
---

*Propuesta de arquitectura del grupo de estudio ANSI/SPARC (1977) para los DBMS*

Nivel conceptual → Esquema conceptual → descripción de la BD sin considerar el DBMS

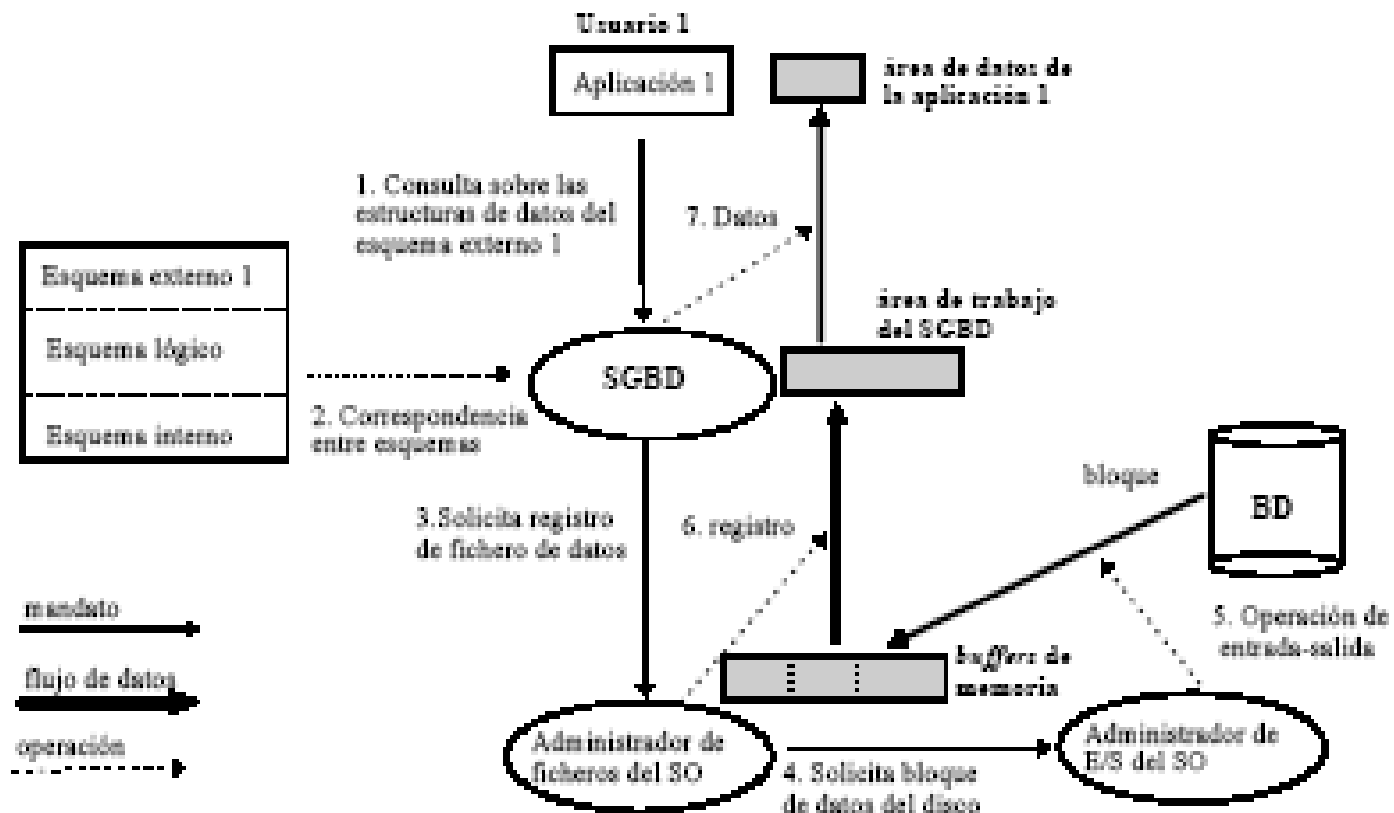
Nivel interno → Esquema interno → descripción de la BD en términos de su representación física

Nivel externo → Esquema externo → descripción de las vistas parciales de la BD para cada grupo de usuarios



# BASES DE DATOS I

## ESQUEMA DE ACCESO DEL SGBD A LOS DATOS





# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***INDEPENDENCIA DE DATOS***

---

Propiedad que asegura que los programas de aplicación sean independientes o inmunes a los cambios realizados en **datos que no usan** o en **detalles de representación física de los datos a los que acceden**.

*Independencia lógica* entre el esquema lógico/conceptual y los externos:  
– Los esquemas externos y los programas de aplicación no deben ser afectados por alteraciones del esquema lógico/conceptual sobre datos que no usan.

*Independencia física* entre el esquema interno y el lógico/conceptual:  
– el esquema lógico/conceptual no debe verse afectado por cambios en el esquema interno referentes a la implementación de las estructuras de datos, modos de acceso, tamaños de páginas, caminos de acceso, etc.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***SGBD / DBMS: SISTEMA DE GESTIÓN DE LA BASE DE DATOS***

---

El ***SGBD/DBMS*** es una aplicación de alta complejidad que:

- Permite a los usuarios definir, crear la base de datos mediante el ***lenguaje de definición de datos (DDL)***.

- Permite la inserción, actualización, eliminación y consulta de datos mediante el ***lenguaje de manejo de datos (DML)***.

Hay dos tipos de lenguajes de manejo de datos:

Operativos ('Procedurales').      No operativos (declarativos) (SQL)

- Proporciona un acceso **controlado** a la base de datos mediante:

- sistema de **seguridad**.
- sistema de **integridad**.
- sistema de **control de concurrencia**.
- sistema de **control de recuperación**.
- **catálogo o diccionario de datos (metadatos)**.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***SGBD / DBMS: SISTEMA DE GESTIÓN DE LA BASE DE DATOS (cont)***

---

- Permite definir ***subesquemas o vistas***: cada usuario “ve” la parte de la base de datos que necesita.
  - . Proporcionan un nivel de **seguridad** adicional.
  - . Proporcionan un mecanismo para que los usuarios vean los datos en el **formato** deseado.
  - . Representan una **imagen consistente y permanente** de la base de datos, aunque ésta cambie su estructura.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***FUNCIONES DE LOS DBMS***

---

1. Permitir a los usuarios **almacenar** datos, **acceder** a ellos y **actualizarlos**, ocultando su estructura física.
2. Proporcionar un **catálogo** accesible por los usuarios.
3. Proporcionar un mecanismo que garantice el **procesamiento de las transacciones**.
4. Proporcionar un mecanismo que realice el **control de la concurrencia**.
5. Proporcionar un mecanismo para **recuperación ante fallos**.
6. Proporcionar un mecanismo de **seguridad**.
7. Integrarse con algún software de **comunicación**.
8. Encargarse de mantener las **reglas de integridad**.
9. Encargarse de mantener la **independencia** entre los programas y la estructura de la base de datos.
10. Proporcionar herramientas para **administrar** la base de datos.

# INTRODUCCIÓN A LAS BASES DE DATOS

## SISTEMAS DE BASE DE DATOS

---

Mantiene ***tres formas de ver los datos (en concordancia con los niveles)***:

-El **esquema de la base de datos**: visión lógica global.

diseñador de la base de datos  
administrador de la base de datos

-El **subesquema o vista**: visión lógica de los datos relacionados con una aplicación.

programadores de aplicaciones  
usuarios finales

-El **esquema físico**: el modo en que los datos están almacenados.

programadores de sistemas  
creadores de DBMS



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***VENTAJAS DE LOS SISTEMAS DE BASES DE DATOS***

---

### **Por la integración de datos:**

- . Se reduce la redundancia de datos, por lo tanto, se ocupa menos espacio en disco.
- . Se pueden evitar los problemas de inconsistencia de datos.
- . Se puede extraer información adicional sobre los datos.
- . Se pueden compartir los datos.
- . Se facilita el mantenimiento de estándares.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***VENTAJAS DE LOS SISTEMAS DE BASES DE DATOS***

---

### **Por las funciones del DBMS:**

- . Se reduce el problema de la integridad de los datos mediante la especificación de reglas de integridad.
- . Se mejora la seguridad de la BD frente a accesos no autorizados.
- . Se pueden consultar los datos sin tener que programar aplicaciones.
- . Se mejora la productividad de los programadores.
- . Proporciona independencia de datos lógica – física.
- . Se aumenta la concurrencia.
- . Mejoran los servicios de copias de seguridad y de recuperación.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***INCONVENIENTES DE LOS SISTEMAS DE BASES DE DATOS***

---

- Los DBMSs son conjuntos de programas muy complejos y muy extensos.
- Son caros.
- Frecuentemente se requiere adquirir equipamiento adicional.
- Si se pasa de un sistema de archivos a un sistema de bases de datos, el costo de la migración es elevado.
- Algunas prestaciones pueden disminuir su desempeño.
- Son más vulnerables a los fallos.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***MODELOS DE DATOS***

---

Un ***modelo de datos*** está formado por:

- Un **conjunto de conceptos** para definir la estructura de la base de datos:

- **Datos y Relaciones** entre datos →

- SENTENCIAS del lenguaje (DDL) para creación/modificación de la estructura/supresión de tablas

- Un **conjunto de conceptos** para definir vínculos significativos (semánticos) entre los datos y relaciones:

- **Restricciones** sobre datos y relaciones.

- SENTENCIAS del lenguaje (DDL) para creación, modificación, supresión de restricciones

- Un **conjunto de operaciones** para realizar consultas y actualizaciones de datos.

- SENTENCIAS del lenguaje (DML) para consultar, insertar, borrar, modificar tuplas



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***LENGUAJES DE LOS DBMS***

---

- . ***Lenguaje de Definición de Datos (DDL):***
  - Definición del esquema de la base de datos.
  - Definición de vistas.
  - Definición de las estructuras de almacenamiento.
- . ***Lenguaje de Manejo de Datos (DML):***
  - Leer datos.
  - Actualizar datos.
- . ***Lenguajes de Cuarta Generación (4GL):***
  - Generadores de formularios.
  - Generadores de informes.
  - Generadores de gráficos.
  - Generadores de aplicaciones.

# INTRODUCCIÓN A LAS BASES DE DATOS

## REGLAS DE INTEGRIDAD

---

**Regla de integridad** : restricción que debe cumplirse sobre una BD en todos sus estados.

**Reglas inherentes al modelo relacional:**

**Regla de integridad de entidades** → claves primarias

**Regla de integridad referencial** → claves extranjeras

**Reglas de negocio** : concepto general que abarca reglas de integridad específicas de cada universo de discurso (plasmadas en la BD)

**Restricciones de dominios** : al definir cada atributo sobre un dominio, se impone una restricción sobre el conjunto de valores permitidos (***nulidad, unicidad***)



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***REGLA DE INTEGRIDAD DE ENTIDADES***

---

"Ninguno de los atributos que componen la clave primaria puede ser nulo."

***Superclave***: conjunto de atributos que identifica de modo único las tuplas de una relación.

***Clave candidata***: superclave en la que ninguno de sus subconjuntos es una superclave de la relación.

***Clave***: Debe satisfacer:

***Unicidad***

***Irreducibilidad (minimalidad)***

### **Observaciones:**

La regla se aplica a las relaciones base (tablas base en la BD)

La regla se aplica sólo a la clave primaria (no a las claves alternativas).



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***REGLA DE INTEGRIDAD REFERENCIAL***

---

"Si en una relación hay alguna clave extranjera, sus valores deben coincidir con valores de la clave primaria a la que hace referencia, o bien, deben ser nulos."

Claves extranjeras de más de un atributos pueden ser parcial o totalmente nulas → diferentes formas de "matching"

Si estando en un estado legal, una operación sobre los datos conduce a un estado ilegal → dos estrategias:

**Rechazar** la operación (*restrict / no action*).

**Aceptar** la operación y **Reparar** realizando operaciones compensatorias adicionales que conduzcan a un estado legal (*cascade / set null / set default*).

# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***SQL***

---

- SQL es un lenguaje para la definición y manipulación de datos, desarrollado inicialmente en los laboratorios de investigación de IBM en California
- Se transformó en un estándar en 1986 (SQL-1986)
- La última versión desarrollada es SQL:1999 (también conocida como SQL3), e incluye características orientadas a objetos
- El estándar previo es SQL-92 (SQL2)



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***SQL***

---

- Componentes
  - DDL (definición de esquemas, relaciones, índices ...)
  - DML (manipulación de datos: inserciones, borrados, modificaciones, consultas)
  - Manipulación de datos empotrada (permite ser usado desde otro lenguaje anfitrión)



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***SQL permite:***

---

- Crear y modificar esquemas, tablas, índices.
- Definir dominios para cada atributo de las relaciones.
- Forzar restricciones de integridad: de dominio, de clave primaria, referenciales, de semántica compleja y/o arbitraria.
- Definir acciones ante violaciones de restricciones.
- Consultar tuplas mediante una combinación de álgebra y cálculo relacional.
- Definir vistas de una relación o combinación de ellas.
- Insertar, borrar y actualizar tuplas en una relación.



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***CREACIÓN DE TABLAS***

---

- EL SQL se basa en el modelo relacional: implementa las relaciones como tablas, los atributos como columnas y las tuplas como filas de las tablas.
- Para definir una tabla en SQL se usa la sentencia **CREATE TABLE**:



# ***INTRODUCCIÓN A LAS BASES DE DATOS***

## ***TIPOS DE DATOS***

---

- Los tipos de datos dependen de la implementación de SQL para cada sistema
  - dialectos
  - adhesión al estándar SQL
- Generalmente se tienen:
  - int: enteros.
  - float: números decimales de punto flotante.
  - datetime: tipo de dato para fechas.
  - money: tipo de dato especial para cantidades contables.
  - char(n): cadena de caracteres de largo fijo.
  - varchar(n): cadena de caracteres de largo variable.
  - binary: para tipos de datos en formato binario (archivos)