

# **Gestión de Bases de Datos**



# Gestión de Bases de Datos

*Rafael Núñez*





Gestión de Bases de Datos  
Thema: UN Bases de datos  
BISAC -COM021000 Database  
© Rafael Núñez  
© De la edición: Ra-Ma 2023

MARCAS COMERCIALES. Las designaciones utilizadas por las empresas para distinguir sus productos (hardware, software, sistemas operativos, etc.) suelen ser marcas registradas. RA-MA ha intentado a lo largo de este libro distinguir las marcas comerciales de los términos descriptivos, siguiendo el estilo que utiliza el fabricante, sin intención de infringir la marca y solo en beneficio del propietario de la misma. Los datos de los ejemplos y pantallas son ficticios a no ser que se especifique lo contrario.

RA-MA es marca comercial registrada.

Se ha puesto el máximo empeño en ofrecer al lector una información completa y precisa. Sin embargo, RA-MA Editorial no asume ninguna responsabilidad derivada de su uso ni tampoco de cualquier violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Esta publicación tiene por objeto proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para el editor ninguna forma de asistencia legal, administrativa o de ningún otro tipo. En caso de precisarse asesoría legal u otra forma de ayuda experta, deben buscarse los servicios de un profesional competente.

Reservados todos los derechos de publicación en cualquier idioma.

Según lo dispuesto en el Código Penal vigente, ninguna parte de este libro puede ser reproducida, grabada en sistema de almacenamiento o transmitida en forma alguna ni por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro sin autorización previa y por escrito de RA-MA; su contenido está protegido por la ley vigente, que establece penas de prisión y/o multas a quienes, intencionadamente, reprodujeren o plagieren, en todo o en parte, una obra literaria, artística o científica.

Editado por:  
RA-MA Editorial  
Calle Jarama, 3A, Polígono Industrial Igarza  
28860 PARACUELLOS DE JARAMA, Madrid  
Teléfono: 91 658 42 80  
Fax: 91 662 81 39  
Correo electrónico: [info@grupoeditorialrama.com](mailto:info@grupoeditorialrama.com)  
Internet: [www.ra-ma.es](http://www.ra-ma.es) y [www.ra-ma.com](http://www.ra-ma.com)  
ISBN impreso: 978-84-1985-747-7  
ISBN ePub: 978-84-19857-48-4  
Depósito legal: M-18112-2023  
Maquetación: Antonio García Tomé  
Diseño de portada: Antonio García Tomé  
Filmación e impresión: Safekat  
Impreso en España en julio de 2023

---

# ÍNDICE

<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>11</b>
1.1 ORIGEN Y EVOLUCIÓN DE LAS BASES DE DATOS .....	11
1.2 CONCEPTO DE BASE DE DATOS. OBJETIVOS .....	13
1.2.1 Sistemas orientados al Proceso .....	13
1.3 INDEPENDENCIA DE LOS DATOS. ARQUITECTURA .....	16
1.3.1 Tipos de independencia .....	17
1.4 SISTEMA DE GESTIÓN DE BASES DE DATOS .....	20
1.5 BENEFICIOS DE LAS BASES DE DATOS .....	25
1.6 INCONVENIENTES DE LAS BASES DE DATOS .....	25
<b>CAPÍTULO 2. MODELOS DE DATOS .....</b>	<b>27</b>
2.1 MODELOS DEL ANÁLISIS (I).....	27
2.2 OBJETIVOS DE LOS MODELOS DE DATOS.....	28
2.3 ESQUEMA .....	29
2.4 ESQUEMA VS. EJEMPLAR .....	29
2.5 COMPONENTES DE UN MODELO DE DATOS.....	30
2.5.1 Estática .....	30
2.5.2 Dinámica .....	31
2.6 TIPOS DE MODELOS DE DATOS.....	33
2.6.1 Primitivos .....	33
2.6.2 Clásicos .....	33
2.7 DIFERENCIAS ENTRE LOS MODELOS SEMÁNTICOS Y LOS CLÁSICOS.....	34

2.7.1	Semánticos .....	34
2.7.2	Clásicos .....	34
2.7.3	Modelo Jerárquico.....	34
2.7.4	Modelo en Red CODASYL.....	36
2.8	CARACTERIZACIÓN DE LOS LENGUAJES DE DATOS .....	38
<b>CAPÍTULO 3. MODELO ENTIDAD / INTERRELACIÓN .....</b>		<b>39</b>
3.1	PRESENTACIÓN DEL MODELO .....	39
3.2	ELEMENTOS DEL MODELO .....	39
3.2.1	Entidad .....	40
3.2.2	Interrelación .....	41
3.2.3	Dominio.....	43
3.2.4	Atributo .....	43
3.3	RESTRICCIONES INHERENTES.....	44
3.3.1	Restricciones de integridad estática .....	45
3.4	IDENTIFICADORES.....	45
3.4.1	Cardinalidades de atributos .....	46
3.4.2	Cardinalidad .....	47
3.4.3	Dependencia en existencia y en identificación.....	48
3.5	CONTROL DE REDUNDANCIA .....	49
3.6	INTERRELACIONES N-ARIAS.....	50
3.6.1	Cardinalidad en interrelaciones n-arias .....	51
3.7	GENERALIZACIÓN .....	54
3.7.1	Características .....	54
3.8	EL TIEMPO EN EL MODELO E/R.....	57
<b>CAPÍTULO 4. DISEÑO DE BASES DE DATOS RELACIONALES: EL MODELO RELACIONAL .....</b>		<b>59</b>
	OBJETIVOS .....	59
4.1	INTRODUCCIÓN .....	59
4.1.1	Reseña histórica.....	61
4.2	ELEMENTOS BÁSICOS.....	62
4.2.1	Dominios .....	63
4.2.2	Elementos básicos: relaciones.....	65
4.3	CLASES DE RELACIONES .....	67
4.3.1	Relaciones derivadas .....	67
4.4	CLAVES .....	68
4.4.1	Claves Candidatas .....	68
4.4.2	Claves Ajenas .....	69
4.5	RESTRICCIONES.....	71
4.5.1	Inherentes .....	71
4.5.2	Semánticas.....	72
4.6	ESQUEMAS RELACIONALES.....	77

4.7	SGBDR .....	77
4.7.1	El MR y la arquitectura ANSI .....	77
4.7.2	Reglas de Codd.....	78
4.7.3	Tratamiento de valores nulos.....	80

## **CAPÍTULO 5. EL MODELO RELACIONAL. ESTÁTICA.**

### **TRANSFORMACIÓN DE ESQUEMAS E/R A ESQUEMAS RELACIONALES..... 83**

5.1	MODELO LÓGICO DE DATOS. OBTENCIÓN MODELO LÓGICO DE DATOS (MLD) A PARTIR DEL MODELO CONCEPTUAL DE DATOS (MCD).....	83
5.1.1	Etapas del diseño lógico .....	83
5.1.2	Transformación del esquema conceptual al lógico estándar .....	84
5.1.3	Reglas de transformación .....	86
5.2	DEFINICIÓN DE OBJETOS EN EL MODELO RELACIONAL. INTRODUCCIÓN AL DDL DEL SQL.....	97
5.2.1	Creación de dominios.....	97
5.2.2	Tipos de datos.....	97
5.2.3	Tipos de datos de usuario: Sentencia CREATE DATATYPE.....	98
5.2.4	Creación de relaciones (Tablas): CREATE TABLE.....	99
5.2.5	Descripción.....	100
5.2.6	Restricciones de tabla.....	100
5.2.7	Restricciones de columna.....	101
5.2.8	Restricciones de integridad .....	101
5.2.9	Creación de índices: CREATE INDEX.....	103
5.2.10	Modificación de la estructura de una tabla: ALTER TABLE.....	103
5.2.11	Borrado de objetos: DROP.....	103
5.2.12	Desencadenadores (Triggers).....	104

### **CAPÍTULO 6. DISEÑO EN EL MODELO RELACIONAL. TEORÍA DE LA NORMALIZACIÓN..... 109**

6.1	INTRODUCCIÓN .....	109
6.2	TEORÍA DE LA NORMALIZACIÓN.....	112
6.2.1	La aproximación por descomposición.....	112
6.2.2	Operaciones básicas sobre relaciones.....	112
6.3	FORMAS NORMALES .....	115
6.3.1	Normalización, enfoque intuitivo.....	115
6.3.2	Concepto de dependencia funcional.....	117
6.3.3	Esquemas de relación .....	118
6.3.4	Propiedades de las dependencias funcionales .....	118
6.3.5	Tipos de dependencias funcionales .....	119
6.3.6	Diagrama de dependencias funcionales .....	120
6.3.7	Claves .....	120
6.3.8	Algoritmo de Ullman para el cálculo del cierre de un descriptor .....	122
6.3.9	Cálculo de Claves.....	123

6.4	NORMALIZACIÓN BASADA EN DEPENDENCIAS FUNCIONALES .....	125
6.4.1	Segunda forma normal (2FN) .....	125
6.4.2	Tercera forma normal (3FN) .....	126
6.4.3	Forma normal de Boyce y Codd (FNBC) .....	126
6.5	OTRAS DEPENDENCIAS Y FORMAS NORMALES .....	127
6.5.1	Dependencias multivaluadas y 4FN .....	127
6.5.2	Definición de dependencia multivaluada .....	129
6.5.3	Definición de 4FN .....	130
6.5.4	Dependencias de reuniones y 5FN .....	132
6.5.5	Definición de dependencia de reunión .....	136
6.5.6	Ejemplo de dependencia de reunión .....	136
6.5.7	Dependencias multivaluadas y dependencias de reunión .....	137
6.5.8	Definición de 5FN .....	137
6.5.9	Ejemplos y observaciones .....	137
6.6	BIBLIOGRAFÍA .....	138
<b>CAPÍTULO 7. DINÁMICA DEL MODELO RELACIONAL .....</b>		<b>139</b>
7.1	INTRODUCCIÓN .....	139
7.2	ÁLGEBRA RELACIONAL .....	140
7.2.1	Operaciones tradicionales de conjuntos .....	140
7.2.2	Operaciones relacionales especiales .....	140
7.2.3	Operadores primitivos .....	141
7.2.4	Operadores derivados .....	141
7.3	OPERACIONES DE ASIGNACIÓN Y DE RENOMBRADO DE ATRIBUTOS .....	142
7.4	OPERADORES PRIMITIVOS .....	144
7.4.1	Unarios .....	144
7.4.2	Restricción ( $\sigma$ ) .....	145
7.4.3	Proyección ( $\pi$ ) .....	146
7.4.4	Binarios .....	147
7.4.5	Unión ( $\cup$ ) .....	147
7.4.6	Diferencia ( $-$ ) .....	148
7.4.7	Producto cartesiano ( $\times$ ) .....	149
7.5	PROPIEDADES DE LOS OPERADORES PRIMITIVOS .....	150
7.6	OPERADORES DERIVADOS .....	151
7.6.1	Reunión theta ( $\theta$ ) .....	152
7.6.2	Reunión natural .....	152
7.6.3	Intersección ( $\cap$ ) .....	152
7.6.4	División ( $:$ ) .....	153
7.7	RESUMEN DE OPERADORES .....	155
7.8	REFERENCIAS Y BIBLIOGRAFÍA .....	156



<b>CAPÍTULO 8. TABLAS UTILIZADAS EN LOS EJEMPLOS.....</b>	<b>157</b>
8.1 TABLAS UTILIZADAS EN LOS EJEMPLOS.....	157
8.1.1 Sentencias de creación de las tablas.....	157
8.1.2 Contenido de las tablas.....	158
8.2 SQL.....	159
8.2.1 Formato básico de la sentencia SELECT.....	159
8.2.2 Eliminación de filas repetidas.....	164
8.2.3 Expresiones.....	166
8.2.4 Predicados.....	167
8.2.5 Sentencias subordinadas.....	170
8.2.6 Predicados propios de SQL.....	173
8.2.7 Funciones de columnas.....	182
8.2.8 Funciones colectivas.....	182
8.2.9 Reglas y formatos de las funciones colectivas.....	183
8.2.10 Consultas con agrupamiento de filas.....	186
8.2.11 Sentencia Select con agrupamiento de filas.....	189
8.2.12 Consultas sobre varias tablas.....	190
8.2.13 Operación de reunión (Join).....	191
8.2.14 Sentencias subordinadas.....	194
8.2.15 Consultas correlacionadas.....	194
8.3 EJERCICIOS.....	197
8.3.1 Composición de consultas.....	199
8.3.2 Creación de vistas.....	201
8.4 INSERCIÓN, MODIFICACIÓN Y BORRADO DE INFORMACIÓN.....	201
8.4.1 Inserción, modificación y borrado de información en vistas.....	204
8.5 SEGURIDAD.....	206
8.5.1 Sentencias GRANT y REVOKE.....	206
<b>CAPÍTULO 9. PROCESAMIENTO DE TRANSACCIONES.....</b>	<b>207</b>
9.1 EL PROBLEMA.....	207
9.1.1 Concepto.....	207
9.1.2 Propiedades de las transacciones.....	207
9.2 CONTROL DE LAS TRANSACCIONES.....	210
9.2.1 Definición de transacción en SQL.....	210
9.2.2 Inicio de transacciones.....	211
9.2.3 Finalización de transacciones.....	211
9.3 MODELOS DE TRANSACCIÓN.....	212
9.3.1 El modelo de transacción ANSI/ISO.....	212
9.3.2 Otros modelos de transacciones.....	213
9.4 TRANSACCIONES Y PROCESAMIENTO MULTIUSUARIO.....	215
9.5 PLANIFICACIONES.....	215
9.5.1 Planificaciones en serie y en paralelo.....	215
9.5.2 Cuatro problemas de la concurrencia.....	219

---

9.6	GARANTÍA DE CONSISTENCIA.....	222
9.6.1	Recuperabilidad.....	222
9.6.2	Planificaciones recuperables.....	223
9.6.3	Planificaciones que evitan abortos en cascada .....	224
9.6.4	Conflicto en planificaciones serializables.....	225
9.6.5	Serializabilidad de vistas .....	226
9.6.6	Pruebas de serializabilidad .....	227
9.7	CONTROL DE CONCURRENCIA .....	229
9.7.1	Cerramiento o bloqueo (Locking).....	229
9.7.2	Control de concurrencia optimista y pesimista .....	230
9.8	CERRAMIENTO (LOCKING) .....	230
9.8.1	Granularidad de los bloqueos.....	231
9.8.2	Cierres compartidos y exclusivos.....	233
9.8.3	Interbloqueos .....	234
9.8.4	Técnicas avanzadas de cerramiento .....	235
9.8.5	Cerramiento explícito.....	236
9.8.6	Niveles de aislamiento (Niveles de Consistencia) .....	237
9.8.7	Parámetros de cerramiento .....	240
9.9	PROTOCOLOS BASADOS EN MARCAS TEMPORALES .....	241
9.9.1	Hora de entrada .....	241
9.9.2	Protocolo de ordenación por hora de entrada.....	241
9.10	LAS OPERACIONES INSERTAR Y BORRAR .....	243
9.10.1	Borrado.....	243
9.10.2	Inserción .....	243
9.11	RECUPERACIÓN .....	244
9.11.1	Clasificación de fallos .....	244
9.11.2	La jerarquía de almacenamiento.....	244
9.11.3	Recuperación basada en bitácora .....	245
9.11.4	Gestión de registros intermedios .....	250
9.11.5	Puntos de verificación .....	251
9.11.6	La bitácora.....	252

# 1

---

## INTRODUCCIÓN

---

### 1.1 ORIGEN Y EVOLUCIÓN DE LAS BASES DE DATOS

---

#### Primera generación (años 50)

- El fichero no existe, los datos solo existen dentro de los programas.
- Nuevos soportes.
  - Cinta perforada y magnética. Acceso secuencial. Posibilidad de búsqueda de información.
- Acceso a ficheros: lenguaje máquina y ensamblador.

#### Segunda generación (años 60)

- Diálogo interactivo con la máquina.
- Disco magnético acceso directo (finales 50).
- Organización secuencial indexada (ISAM).
- Sistemas de ficheros (sistemas orientados al proceso).

#### Segunda generación (años 60)

- Sistemas de ficheros.
  - Asociación estática de los ficheros a los programas de forma individual.
    - Ficheros integrados en la aplicación y el hardware.
    - Ficheros a la medida de la aplicación.
    - Formatos de ficheros heterogéneos.
    - Redundancia y problemas de compartición.

### Tercera generación (años 70, prerrelacional)

- Unificación de la información sin perder la perspectiva de usuarios.
- Distinción entre estructura lógica global y vista de usuario.
  - Arquitectura a dos niveles.
- Distinción entre significado y valor almacenado.
- Evolución de las bases de datos.
  - En aplicaciones técnicas/militares: hechas a medida.
  - Años 60: primer software para un conjunto de aplicaciones: BOMP, CFMS, etc.
  - 1962-64: introducción de estructuras de **cadena**s y **anillos** por Bachman (IDS → Codasyl).
  - 1969 (Desde 1965): IMS/1 de IBM, inicialmente diseñada para el proyecto Apollo.
  - 1968: CODASYL a partir del modelo de Bachman establece el concepto de conjunto (**set**) → **Modelo RED**.

### Cuarta generación (años 80, relacional)

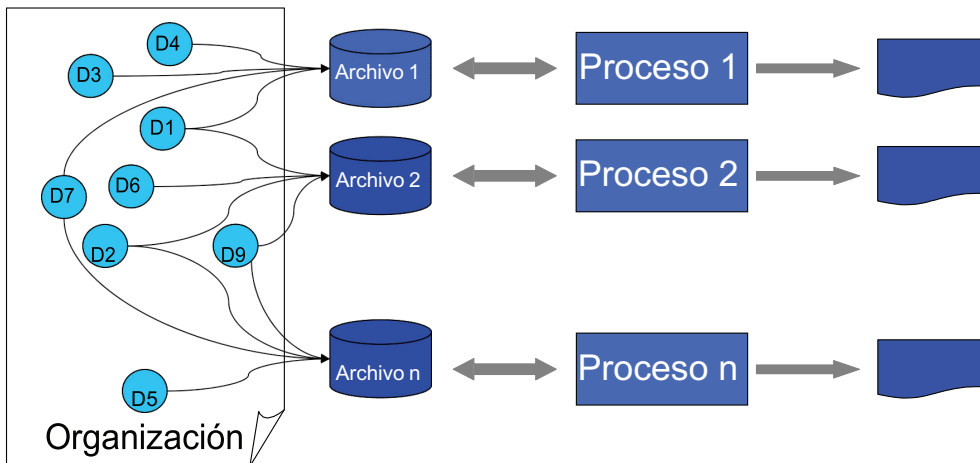
- Nuevos productos: sistemas de Bases de Datos.
- Control centralizado → evita y/o controla la redundancia.
- Clara distinción entre el modelo lógico y el físico.
  - Modelo Relacional **solo** tiene representación lógica.
- Alto grado de independencia de datos.
- Almacenamiento transparente al usuario.
- Lenguajes más potentes (*Qué* en lugar de *Cómo*).
- Modelo relacional (Codd, 1969-1970).
  - Basado en el álgebra y la teoría de conjuntos y relaciones binarias.
  - Prototipos.
    - INGRES Universidad de Berkeley (1973-75).
    - SYSTEM-R de IBM (1974-77).
  - Sistemas comerciales.
    - INGRES de RTI (1980)
    - SQL/DS de IBM (1981).
    - ORACLE de RSI (1981).
    - DB2 de IBM (1983).

### Quinta generación (años 90, postrelacional)

- Bases de datos deductivas.
- Bases de datos orientadas a objetos.
- Objeto-Relacional.
- Bases de datos activas.
- Datos no estructurados.
- Bases de datos NoSQL.

## 1.2 CONCEPTO DE BASE DE DATOS. OBJETIVOS

### 1.2.1 Sistemas orientados al Proceso



### Inconvenientes

- Redundancia.
  - Desaprovechamiento de espacio de almacenamiento.
  - Excesivo mantenimiento.
  - Inconsistencia de los datos.
- Poca flexibilidad frente a cambios.
- Baja productividad.
- Limitación de recursos compartidos.
- Medidas de seguridad difíciles.
- Dificultad para hacer cumplir las normas de la empresa.

- Aislamiento de los datos.
- Atomicidad difícil de conseguir.
- Anomalías de accesos concurrentes.
- Imposible responder a demandas inesperadas de información.
  - Dificultad de acceso a los datos (los datos están, pero no pueden ser convenientemente accedidos).
- Dependencia de datos y aplicaciones.

### Objetivos de las BBDD

- Independencia de datos (Flexibilidad).
  - Los cambios en las aplicaciones no deben imponer un nuevo diseño y viceversa.
- Coste mínimo.
  - Adaptación rápida y con coste mínimo a las nuevas características de la empresa.
- Datos compartidos.
  - Permite una mayor disponibilidad de los datos.
- Versatilidad en la representación de los datos.
  - Distintos usuarios quieren ver de forma distinta los datos.

### Objetivos

- Capacidad de búsqueda.
  - Rapidez y flexibilidad en la exploración de la base de datos.
- Consistencia y mínima redundancia.
  - Evitar repetición innecesaria de datos y la incoherencia entre estos.
- Integridad.
  - Garantizar la exactitud y veracidad de los datos.
- Tolerancia a fallos y seguridad.
  - Protección de los datos ante accesos no autorizados y fallos.

### Definiciones (I)

- *Colección de datos interrelacionados.* (Elmasri y Navathe, 1989).
- *Colección **no redundante** de datos que son **compartidos** por diferentes programas de aplicación.* (Howe, 1983).
- *Conjunto de datos de la empresa memorizado en un ordenador, que es utilizado por numerosas personas y **cuya organización está regida por un modelo de datos.*** (Flory, 1982).

## Modelo de Datos

- Conjunto de herramientas conceptuales para describir los datos, las relaciones entre ellos, su semántica y las restricciones de consistencia que les afectan.
- Tipos.
  - Semánticos (Conceptuales) (No tienen representación en máquina).
  - Lógicos (Convencionales) (Sí tienen representación en máquina).
    - Red (CODASYL).
    - Jerárquico.
    - Relacional.

## Esquemas y Ejemplares

- El resultado de representar una información en un determinado modelo de datos es un esquema de datos en ese modelo.
  - Subesquemas.
  - Esquema lógico.
  - Esquema físico.
- Un ejemplar del esquema es la información que el esquema tiene en un momento determinado.

## Definiciones (II)

- *Una base de datos es un conjunto de información almacenada en memoria auxiliar que permite acceso directo, y un **conjunto de programas que manipulan esos datos**.*
- *Una base de datos es un conjunto exhaustivo, no redundante de datos estructurados, **organizados independientemente de su utilización y su implementación en máquina**, accesibles en tiempo real y compatibles por usuarios concurrentes que tienen necesidad de información diferente y no predecible en el tiempo.*

## Definiciones (y III)

- *“Colección o depósito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes a estas, y su definición y descripción, únicas para cada tipo de dato,*

*han de estar almacenados junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos.”* (de Miguel y Piattini, 1999).

### 1.3 INDEPENDENCIA DE LOS DATOS. ARQUITECTURA

---

Garantía de consistencia frente a accesos concurrentes y tolerancia a fallos.

#### Concepto de Transacción

- Conjunto de operaciones que deben realizarse de forma atómica (todas o ninguna).
- Garantizar la consistencia de los datos.
- Debe estar aislada de otras transacciones.
- Debe perdurar en el tiempo, los cambios que produce deben quedar reflejados en la base de datos.
- SQL.
  - COMMIT/ROLLBACK.

Implica la separación lógica y física de los datos en la base de datos.

- Influencia en la arquitectura del sistema.
- Grado de dependencia.
- ANSI: *“La independencia de los datos es la capacidad de un sistema para permitir que las referencias a los datos almacenados, especialmente en los programas y en sus descriptores de los datos, estén aislados de los cambios y de los diferentes usos en el entorno de los datos, como pueden ser la forma de almacenar dichos datos, el modo de compartirlos con otros programas y cómo se reorganizan para mejorar el rendimiento del sistema de Bases de Datos.”*

#### Independencia de datos es (I)

- La capacidad de un sistema de gestión de bases de datos para permitir que las referencias a los datos almacenados, especialmente en los programas



y sus descripciones de los datos, estén aisladas de los cambios y de los diferentes usos en el entorno de los datos, como pueden ser:

- La forma de almacenar dichos datos.
- El modo de compartirlos con otros programas.
- Cómo se reorganizan para mejorar el rendimiento del sistema de bases de datos.

## Independencia de datos es (II)

- La inmunidad de las aplicaciones ante cambios de la estructura de almacenamiento y de los métodos de acceso.

### 1.3.1 Tipos de independencia

- De descripción.
  - Separación de la definición de datos a nivel físico y lógico.
- De manipulación.
  - Independencia respecto a los caminos de acceso y soportes físicos.

#### Independencia lógica

*Capacidad para modificar el esquema **lógico** sin provocar que los programas de aplicación tengan que reescribirse.*

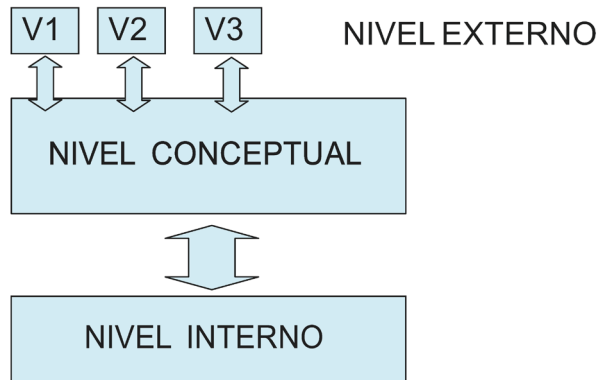
#### Independencia física

*Capacidad para modificar el esquema **físico** sin provocar que los programas de aplicación tengan que reescribirse.*

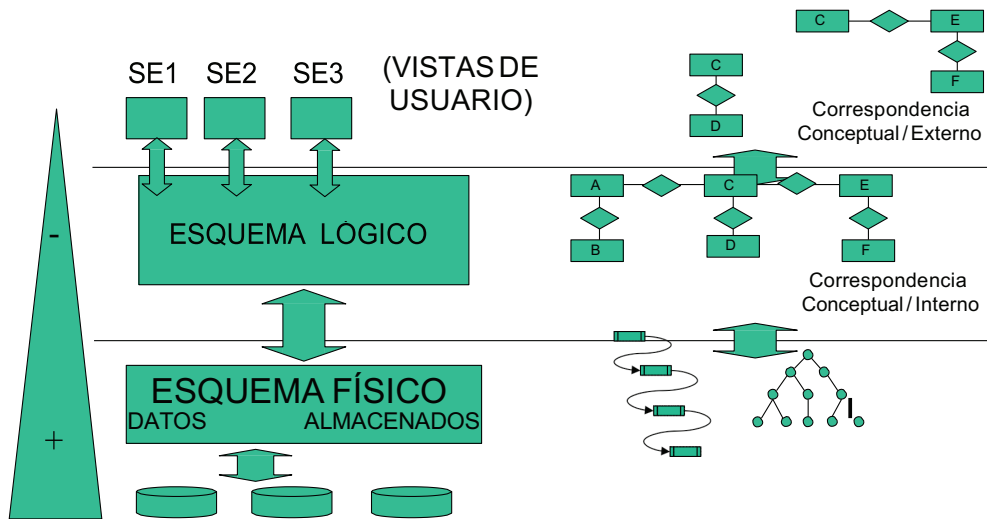
### ¿Cómo se consigue la independencia de datos?

- Arquitectura a tres niveles.
  - Propuesta por el comité ANSI/X3/SPARC (1975).
    - Nivel externo.
    - Nivel conceptual.
    - Nivel interno.

- Asocia a cada nivel un esquema.
  - Esquema externo.
  - Esquema conceptual.
  - Esquema interno.
- Arquitectura a tres niveles.



- **Nivel Externo:** visión que de la base de datos tiene un usuario o aplicación en particular. Habrá tantas vistas de la base de datos como exijan las diferentes aplicaciones. Las vistas se derivan directamente del esquema conceptual, o de otras vistas, y contienen una descripción de los elementos de datos y sus interrelaciones orientadas al usuario o aplicación y de las que se compone la vista. Una misma vista puede ser utilizada por varias aplicaciones.
- **Nivel Conceptual:** contiene el diseño conceptual de la base de datos, que implica el análisis de las necesidades de información de los usuarios y las clases de datos necesarias para satisfacer dichas necesidades. El resultado del diseño conceptual contiene la descripción de todos los datos y las interrelaciones entre ellos, así como las restricciones de integridad y de confidencialidad.
- **Nivel Interno:** en él se define la estructura física de la base de datos: dispositivos de almacenamiento físico, direcciones físicas, estrategias de acceso, relaciones, índices, apuntadores, etc. Es responsabilidad de los diseñadores de la base de datos física. Ningún usuario, en calidad de tal, tiene conocimiento de este nivel.



### Esquema Interno (Nivel Interno)

- Cómo se almacenan los datos. Cómo se accede a ellos.
  - Estructuras de datos complejas a bajo nivel.
  - Es el nivel más próximo al almacenamiento físico.
  - Trata los datos como registros internos (no como bloques de datos).
  - Contiene información sobre:
    - Estructuras de datos usadas.
    - Mecanismo de acceso.
    - Distribución de registros en el espacio lógico.

### Esquema Lógico (Nivel Conceptual)

- Qué datos se almacenan. Cómo están relacionados.
  - Vista **lógica** general de todos los datos.
  - Interrelaciones entre los datos.
  - Uso de un modelo de datos que proporcione un nivel de abstracción de la vista interna.
  - Uso de un lenguaje de definición de datos (DDL).

### Nivel de Vistas (Nivel Externo)

- Solo parte de la base de datos.
  - Comprende las vistas individuales de los usuarios (subesquemas). Muchos usuarios necesitan ver *solo una parte* de la base de datos.
  - Muchos usuarios necesitan vistas *distintas* de los mismos datos.
  - Para simplificar la interacción con el sistema, se define una abstracción en el nivel de vistas.
  - El sistema puede proporcionar muchas vistas de la misma base de datos.

### Interfaces entre niveles

- Los proporciona el SGBD (DBMS).
- Transformaciones entre los esquemas.
- Consultas y actualizaciones.
  - DML (Lenguaje de Manipulación de Datos).
    - Lenguaje interactivo.
    - Precompilador.
    - Extensión del compilador.

### Interfaces entre niveles

- Creación de las estructuras.
  - DDL (Lenguaje de Definición de Datos).
    - DDL interno.
    - DDL conceptual.
    - DDL externo.
    - DDL + DML = DSL (sublenguaje de datos).
    - DML → todos los usuarios.
    - DDL → DBA (Administrador de la base de datos).

## 1.4 SISTEMA DE GESTIÓN DE BASES DE DATOS

---

### Sistemas de Gestión de BBDD (SGBD)

- Propósito: proporcionar una visión **abstracta** de los datos (ocultar los detalles de cómo se almacenan y se mantienen los datos).

- Conjunto de programas que actúa como intermediario entre los usuarios y los datos.
- Recoge las peticiones de los usuarios y responde a ellas.
- Gestión y recuperación eficiente: estructuras de datos complejas.
- El sistema oculta la complejidad para facilitar la interacción con los usuarios a través de los tres niveles de abstracción.

### Usuarios de un SGBD

- Usuarios Informáticos.
  - Diseñadores.
  - Administradores.
  - Analistas y programadores.
- Usuarios Finales.

### Características

- Descripción de la BD exterior a los programas y gestionada por el SGBD.
- Los programas no leen ni escriben directamente sobre la estructura interna de la BDD.
- Gestión de:
  - Control de accesos concurrentes.
  - Control de autorizaciones de acceso.
- Reconstrucción y/o restauración de la BD (tolerancia a fallos).

### El Catálogo o Diccionario de datos

- Capacidad del SGBD de describir los datos que contiene.
- Es una base de datos que contiene información sobre los datos almacenados en la propia base de datos (metadatos).
- Es una base de datos del sistema (no de usuario).
- Debe suministrar documentación única y actualizada de forma que pueda ser usada para obtener información sobre los tipos de datos almacenados y cómo se deben usar.
- Contiene los esquemas y las correspondencias entre ellos (externo / conceptual / interno).

## Componentes de un SGBD

- ▼ **Procesador de I/O.**
  - Directamente asociado al usuario.
  - Toma las órdenes y muestra la respuesta.
- ▼ **Analizador.**
  - Análisis sintáctico de la orden → Diccionario.
  - DDL → Actualización del Diccionario.
  - Lenguaje embebido → Precompilador.
- ▼ **Control de autorizaciones.**
  - Chequeo de autorizaciones → Diccionario.
- ▼ **Obtención de código intermedio.**
  - Componentes de un SGBD.
- ▼ **Procesador de actualizaciones y control de integridad.**
  - Ejecución de la actualización.
  - Control de consistencia.
- ▼ **Procesador de consultas y optimizador.**
  - Transforma la consulta en términos conceptuales.
  - Reformulación para la optimización del acceso.
- ▼ **Generador de código ejecutable.**
  - Secuencia de lecturas y escrituras en disco.
- ▼ **Gestor de transacciones.**
  - Sincronización de los accesos concurrentes.
- ▼ **Gestor de recuperación.**
  - Recuperación del estado de la BDD previo al fallo.
- ▼ **Gestor de datos.**
  - Gestión del hardware.
  - Ejecución de los accesos físicos.

## Estructuras del SGBD por interfaces

Interfaz	Componente SGBD
Usuario	Procesador I/O
Externo / Conceptual	Analizador, precompilador, procesador de actualizaciones y procesador de consultas
Conceptual / Interno	Generador de código, optimizador
Interno / Base de datos	Gestor de transacciones, gestor de datos

## Arquitectura de un SGBD multiusuario

- Arquitectura cliente-servidor.
  - El SGBD actúa como **servidor** proporcionando todo el soporte de los niveles externo, conceptual, e interno.
  - Las aplicaciones que se ejecutan sobre SGBD actúan como **cliente** y las interfaces de consulta que interactúan con el usuario y envían consultas u otros comandos al servidor.
  - **Ej.** Los sistemas relacionales utilizan el lenguaje SQL para representar peticiones del cliente ante el servidor. El servidor las procesa y envía la respuesta al cliente en forma de tabla o relación.

## Funciones del servidor

- Aceptar y procesar las peticiones de los clientes de la BD.
- Comprobar autorizaciones.
- Asegurar que se cumplen las restricciones de integridad.
- Realizar los procesos de consulta/actualización y transmitir la respuesta al cliente.
- Mantener el diccionario de datos.
- Proporcionar acceso concurrente a la BDD, ...

## Tareas de administración

- **La estructura de la base de datos.**
- **Descripción conceptual.** Una vez conocidos los requisitos de información, es preciso realizar el diseño conceptual.
- **Descripción física de los datos.** Encontrar una estructura interna que soporte el esquema lógico y los objetivos de diseño con la máxima eficiencia de los recursos máquina.
- Especificaciones de **vistas** o **subesquemas**.
- **Los estándares.** En cuanto a documentación, metodologías.
- **Procedimientos de explotación y uso.**
- **Aspectos relativos a seguridad, integridad y confidencialidad.**

## Herramientas que utiliza el administrador

- **Lenguajes de definición de datos (DDL)**
- **Utilidades del SGBD** (copias de seguridad, asignación de usuarios, simulación de procesos, documentación, ...).
- **Diccionario de datos.** Contendrá la definición y descripción de todos los datos (metadatos). Ayuda a la gestión de la información como recurso y conseguir la integración de la semántica de forma centralizada.

## Tipos de administradores

- El administrador a nivel de **empresa** (N. Conceptual).
- El administrador de la **base de datos** (N. Interno).
- Los administradores por **aplicaciones** (N. Externo).
- Funciones del administrador de la **BDD** (i).
  - Fijar, a nivel orgánico, la compatibilidad entre trabajos y el orden de ejecución de los trabajos incompatibles.
  - Obtener estadísticas sobre la base de datos.
  - Proporcionar a los operadores del ordenador descripciones completas de lo que hay que hacer para la reconstrucción de las bases de datos.
  - Establecer normas para el uso eficiente de la base de datos.
  - Implementar los requisitos en cuanto a seguridad establecidos por el administrador a nivel de empresa.
- Funciones del administrador de la **BDD** (ii).
  - Escoger los valores de los parámetros físicos para la optimización de los accesos.
  - Proceder a las reestructuraciones de las bases de datos exigidas por los cambios en el esquema interno.
  - Seleccionar y desarrollar programas de servicios relativos a aspectos físicos de las bases de datos.
  - Definir y mantener los esquemas internos, así como la documentación sobre ellos.
- Funciones del administrador por **aplicaciones.**
  - Definir los esquemas externos y mantenerlos.
  - Asignar autorizaciones de acceso a la base de datos a los usuarios de estas.



## 1.5 BENEFICIOS DE LAS BASES DE DATOS

---

Referidas a:

### ➤ LOS DATOS

- Independencia de estos respecto a los tratamientos y viceversa.
- Mejor disponibilidad de los mismos.
- Mayor eficiencia en la recogida, codificación y entrada en el sistema.

### ➤ LOS RESULTADOS

- Mayor coherencia.
- Mayor valor informativo.
- Mejor y más normalizada documentación.

### ➤ LOS USUARIOS

- Acceso más rápido y sencillo de los usuarios finales.
- Más facilidades para compartir los datos por el conjunto de los usuarios.
- Mayor flexibilidad para atender a demandas cambiantes.

## 1.6 INCONVENIENTES DE LAS BASES DE DATOS

---

- Instalación costosa.
- Necesidad de personal especializado.
- Implantación larga y difícil.
- Falta de rentabilidad a corto plazo.
- Escasa estandarización.
- Desfase entre teoría y práctica.



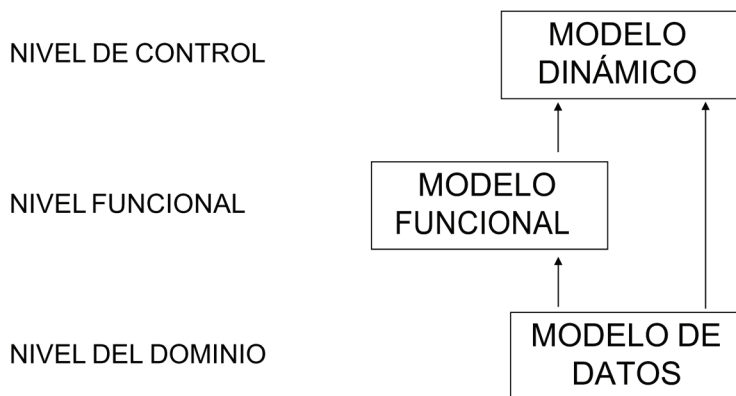
# 2

---

## MODELOS DE DATOS

### 2.1 MODELOS DEL ANÁLISIS (I)

---



### Modelo Funcional

- ¿Qué cosas ocurren, o se desea que ocurran?
- Captura lo que el sistema debe hacer con el conocimiento del dominio, sin considerar cuándo o cómo lo hará.

### Modelo de Control

- ¿Cuándo y a quién le ocurren las cosas?
- Captura el control del sistema en cuanto al tiempo. Controla cuándo ocurren las cosas sin especificar en qué consisten ni cómo son implementados.

### Modelo de Datos

- Conocimiento estático del dominio. Identifica los objetos del dominio y las relaciones y dependencias entre esos objetos.

## 2.2 OBJETIVOS DE LOS MODELOS DE DATOS

---

### Objetivo

Obtención, por medio de una abstracción del mundo real, de un conjunto estructurado de datos y un conjunto de operaciones definidas sobre ellos que permitan satisfacer las necesidades de información de una organización.

### ¿Qué son?

Herramientas que permiten realizar el proceso de abstracción que conduce del mundo real al mundo de los datos.

### Definiciones

- *Conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones. (Dittrich, 1994).*
- *Conjunto de conceptos, reglas y convenciones que permiten describir y manipular los datos de la parcela de un cierto mundo real que deseamos almacenar en la base de datos. (de Miguel, Piattini y Marcos, 1999).*

## 2.3 ESQUEMA

---

*La descripción específica de un determinado minimundo en términos de un modelo de datos se denomina esquema (o esquema de datos) del minimundo. La colección de datos que representan la información acerca del mini mundo constituye la base de datos. (Dittrich, 1994).*

*Representación de un determinado mundo real (universo del discurso) en términos de un modelo de datos. (de Miguel, Piattini y Marcos, 1999).*

## 2.4 ESQUEMA VS. EJEMPLAR

---

Un ejemplar de un esquema son los datos que en un determinado momento se encuentran almacenados en el esquema.

La colección de ejemplares de todos los elementos de un esquema en un momento determinado constituyen un ejemplar del esquema.

Al igual que en los lenguajes de programación existen variables (constituidas por un tipo y un contenido) las cuales tienen, en un momento determinado, un cierto valor, en las bases de datos se debería hablar de *variables de base de datos*, cuyo tipo sería el esquema y su contenido todos los posibles valores del esquema; su valor en un momento determinado sería un ejemplar del esquema.

Nosotros utilizaremos la expresión “base de datos” en el sentido abstracto de todos los posibles ejemplares, y cuando queramos referirnos a su contenido en un cierto momento hablaremos de un ejemplar de la base de datos o de la “*base de datos en el momento i*”

### Clasificación

- ▀ Según el nivel de abstracción de la arquitectura ANSI, tendremos modelos internos, globales o externos.
  - **Externo.**
    - Punto de vista de cada usuario en particular.
  - **Global.**
    - Punto de vista del conjunto de los usuarios (organización).
  - **Interno.**
    - Punto de vista de la máquina.

## 2.5 COMPONENTES DE UN MODELO DE DATOS

---

- Estáticas.
  - Elementos permitidos.
    - Objetos.
    - Asociaciones.
    - Características de los objetos.
    - Dominios.
  - Elementos no permitidos (Restricciones).
    - Inherentes.
    - De integridad o semánticas.
- Dinámicas (Conjunto de operadores).
  - Desde punto de vista estático:
    - Objeto.
    - Relación.
    - Restricción de integridad estática.
  - Desde punto de vista dinámico:
    - Operación.
    - Transacción.
    - Restricción de integridad dinámica.

### 2.5.1 Estática

**Tipo de Objeto:** cualquier “cosa” (persona, concepto, suceso, etc.) con existencia independiente de la que se quiere almacenar información.

- No pueden representarse de forma individual todos los objetos observados. Se representa la *clase* o *tipo* genérico al que pertenecen las instancias u ocurrencias del objeto representado. (*Clasificación*).
- Un *tipo de objeto* en un modelo de datos es el **concepto** que permite representar un conjunto de objetos similares.
- Un tipo de objeto define genéricamente un conjunto de objetos a través de sus *atributos* (información que interesa guardar de esos objetos).

**Tipo de Relación:** asociación entre objetos.

- No se puede representar cada una de las relaciones individuales. Se representan los *tipos de relación*.

- Es el concepto que permite representar un conjunto de relaciones de características similares.
- Al igual que los tipos de objeto, pueden tener *atributos*.
- Diferenciar entre relación (tipo de relación) y ocurrencia de relación (una relación concreta de ese tipo).

**Restricción de integridad estática:** es una propiedad del mundo real, representado para el esquema, que limita sus extensiones validas (ocurrencias permitidas).

Las restricciones forman parte del esquema de la base de datos y deben cumplirse en cualquiera de sus extensiones.

Consisten en:

- Restricciones sobre objetos.
  - Restricciones de valor (dominio de definición).
  - Restricción de valor no-nulo.
- Restricciones sobre relaciones.
  - Restricciones de cardinalidad: limitan la participación de los objetos en las relaciones.

## 2.5.2 Dinámica

- **Operación:** acción elemental (indivisible) sobre un objeto o una relación. Permiten crear, eliminar, modificar o consultar objetos y relaciones.
- **Transacción:** secuencia de operaciones que se considera atómica en lo que se refiere a su ejecución (se ejecutan todas las operaciones o ninguna). Son necesarias porque en ocasiones no es posible representar la evolución del mundo real con operaciones aisladas.
- **Restricción de integridad dinámica:** representan propiedades del mundo real que restringen la posible evolución en el tiempo de la base de datos, limitando la secuencia de extensiones válidas del esquema.

Restricciones de transición: limitan las posibles bases de datos consecutivas.

### ➤ Restricciones semánticas:

- Restricciones **propias**: se especifican al definir el esquema mediante las facilidades que proporciona el lenguaje de definición de datos, almacenándose en el catálogo de la base de datos (no en los programas), por lo que no pueden ser violadas por ninguna aplicación. Cualquier actualización está obligada a respetarlas.
  - **De acción general**: es preciso programar un procedimiento que determine la acción que hay que llevar a cabo.
    - **Procedimientos almacenados**: definidos totalmente de forma procedimental.
    - **Desencadenadores (triggers)**: en los que se formula una condición de forma declarativa. El cumplimiento de la misma “dispara” una acción especificada en forma procedimental.
  - **De acción específica**: la acción (rechazo u otra, predeterminada o elegida mediante opciones) está implícita en la propia restricción.

### Restricciones propias de acción específica

- De **condición general**: se define mediante una condición lógica. La operación a la que afecta será una actualización. No se declara la acción porque este tipo de restricción lleva siempre asociado el rechazo de la operación cuando no se cumple la condición.
  - **De verificación**: la expresión lógica está definida sobre uno o varios atributos de un mismo elemento. (Ej. Cláusula “CHECK”).
  - **De aserción**: similares a las anteriores, pero pueden referirse a más de un elemento del esquema. Tienen existencia por sí mismas y, por lo tanto, nombre (Ej. “CREATE ASSERTION”).
  - **De condición específica o implícitas**: diversas opciones que proporcionan los distintos modelos de datos cuando se definen los elementos de su esquema y que, en realidad, son restricciones. Por ejemplo, en el modelo relacional: PRIMARY KEY, FOREIGN KEY, NOT NULL, etc.



---

## 2.6 TIPOS DE MODELOS DE DATOS

---

- Primitivos (sistemas de ficheros).
- Clásicos.
  - Red.
  - Jerárquico.
  - Relacional.
- Orientados a Objetos.
- Semánticos.

### 2.6.1 Primitivos

- Objetos: registros organizados en ficheros.
- Relaciones: referencias explícitas mediante punteros.
- Lenguajes: totalmente dependientes de la organización física de los ficheros.
- Operadores: las operaciones primitivas sobre ficheros: lectura y escritura de registros.

### 2.6.2 Clásicos

- **Red y Jerárquico:** extensiones de los sistemas de ficheros que buscan una mayor eficiencia en la manipulación de las relaciones entre objetos.
  - Objetos: registros organizados en ficheros.
  - Relaciones: estructuras de datos más complejas que permiten expresar directamente las relaciones entre objetos.
  - Lenguajes: constan de operadores que permiten manejar estas estructuras y que son independientes de su representación física.
- **Relacional:** ruptura con la situación anterior.
  - Objetos y relaciones: se representan mediante *relaciones* (tablas).
  - Lenguajes: totalmente declarativos.
- **Orientados a Objetos:**
  - Limitaciones del Modelo Relacional.
  - Le incorporan al Modelo Relacional conceptos del Modelo Orientado a Objetos con el fin de superar sus limitaciones y dotarle de mayor capacidad expresiva.