

# Sistemas Gestores de bases de datos

## Tema 1: Conceptos de bases de datos

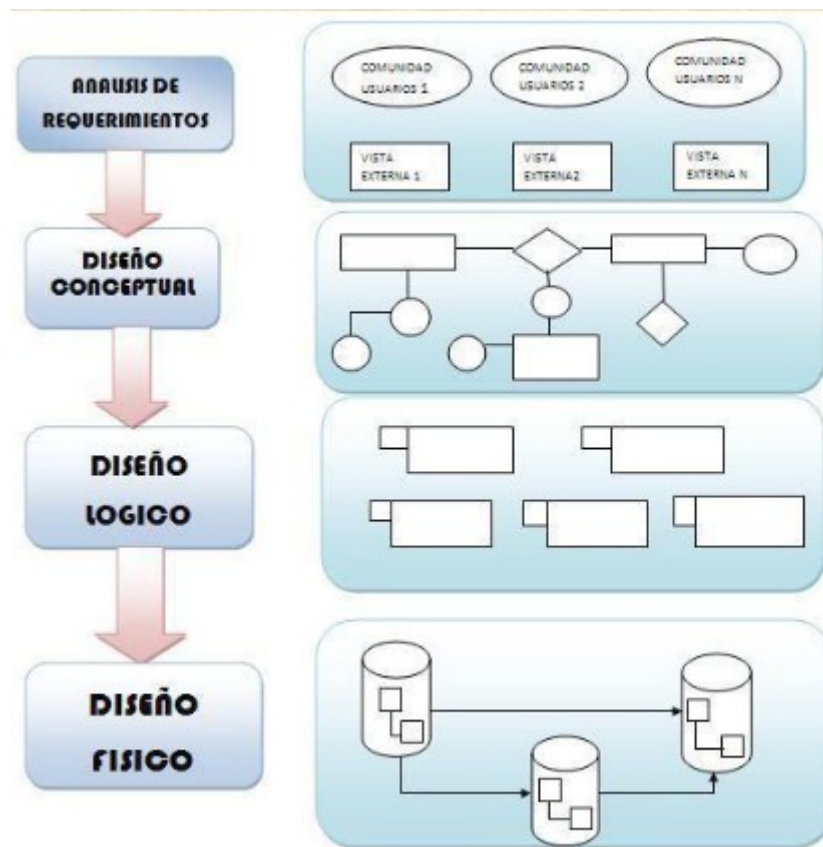
### Índice

1	Introducción. Fases de desarrollo de una BD.....	2
2	Definiciones BD y SGBD.....	3
3	Arquitectura de un SGBD.....	4
4	Funciones de un SGBD.....	6
5	Objetivos y características de un SGBD.....	7
6	Componentes de un SGBD.....	7
7	Usuarios.....	9
8	Tipos de SGBD.....	10
9	SGBD comerciales y libres.....	11
9.1	Tipos de licencias.....	11
9.2	SGBD de código abierto.....	12
9.3	SGBD de código cerrado.....	12
10	Repaso del modelo relacional.....	13
11	Bases de datos NoSQL.....	14
12	Actividades.....	16

# 1 Introducción. Fases de desarrollo de una BD

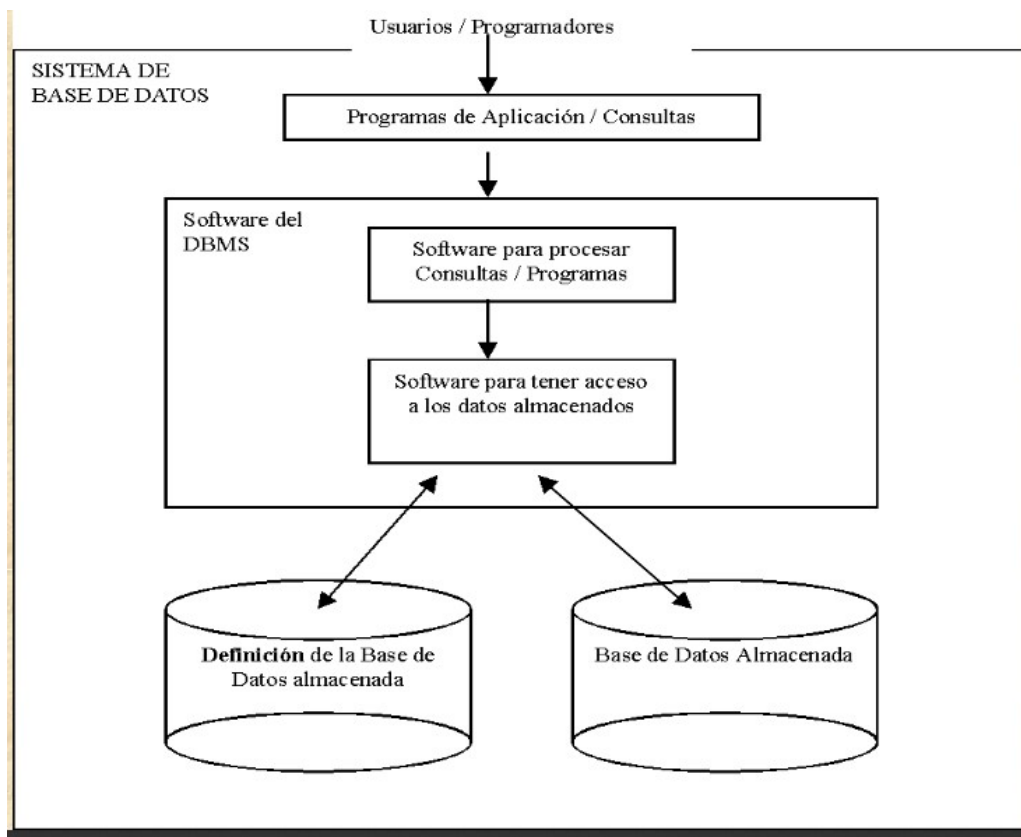
- Un Sistema de Base de Datos es una representación informática de un hecho de la realidad.
- Hay una gran distancia entre la realidad y las estructuras de datos almacenadas en los soporte físicos de un ordenador, luego para llegar a dicha representación requerimos de varias fases.
- Una vez realizado e implantado. Vendría otra nueva fase, a veces, muy erróneamente, no tenida en cuenta: EL MANTENIMIENTO.

**Fases de desarrollo:**



## 2 Definiciones BD y SGBD

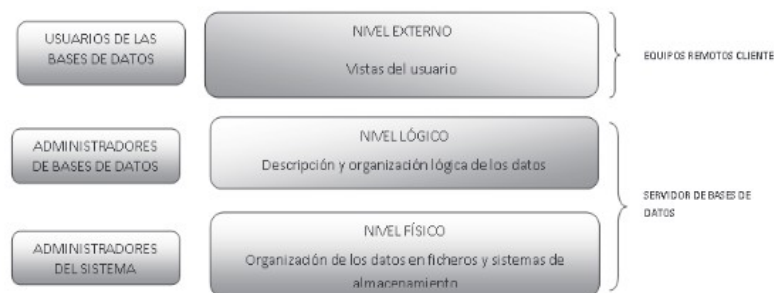
- BD: Conjunto de datos relacionados y organizados con cierta estructura. Información relevante de la empresa.
- SGBD: Aplicación que permite definir crear y mantener la BD. Interfaz entre el Usuario y la BD.
- Sistema BD = BD + SGBD



### 3 Arquitectura de un SGBD

Se basa en la ANSI-SPARC que lo divide en 3 niveles:

- Nivel Interno: El más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. (Archivos, tipos registro, longitud...).(1)
- Nivel Conceptual: Esquema conceptual. (Entidades, relaciones, atributos...)(1)
- Nivel Externo: Es el más cercano a los usuarios. (Vistas) (N)



*Figura 1.1. Arquitectura de Sistemas de Bases de Datos*

**Nivel externo:** Visión parcial de las tablas de la BD según el usuario. Por ejemplo, la vista que se muestra en la Tabla, obtiene el listado de notas de alumnos con los siguientes datos: Curso, Nombre, Nombre de asignatura y Nota

Curso	Nombre	Nombre de asignatura	Nota
1	Ana	Programación en lenguajes estructurados	6
1	Ana	Sistemas informáticos multiusuario y en red	8
2	Rosa	Desa. de aplic. en entornos de 4.ª Generación y H. Case	5
2	Juan	Desa. de aplic. en entornos de 4.ª Generación y H. Case	7
1	Alicia	Programación en lenguajes estructurados	5
1	Alicia	Sistemas informáticos multiusuario y en red	4

**Nivel lógico y conceptual:** Definición de todas las tablas, columnas, restricciones, claves y relaciones. En este ejemplo, disponemos de tres tablas que están relacionadas:

- Tabla ALUMNOS:
  - Columnas: NMatrícula, Nombre, Curso, Dirección, Población.
  - Clave: NMatrícula.



- Además tiene una relación con NOTAS, pues un alumno puede tener notas en varias asignaturas.
- Tabla ASIGNATURAS:
  - Columnas: Codigo, Nombre de asignatura.
  - Clave: Codigo.
  - Está relacionada con NOTAS, pues para una asignatura hay varias notas, tantas como alumnos la cursen.
- Tabla NOTAS:
  - Columnas: NMatricula, CodAsignatura, Nota.
  - Clave NMatricula + CodAsignatura
  - Está relacionada con ALUMNOS y ASIGNATURAS, pues un alumno tiene notas en varias asignaturas, y de una asignatura existen varias notas, tantas como alumnos.

Podemos representar las relaciones de las tablas en el nivel lógico como se muestra en la Figura:



**Nivel interno:** En una BD las tablas se almacenan en archivos de datos. Si hay claves, se crean índices para acceder a los datos de forma más rápida, todo esto contenido en el disco duro, en una o varias pistas y en uno o varios sectores, que sólo el SGBD conoce. Ante una petición, sabe a qué pista, a qué sector, a qué archivo de datos y a qué índices acceder.

#### Que consigue esta arquitectura:

- **Independencia lógica:** la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se podrá modificar el esquema conceptual para ampliar la BD o para reducirla, por ejemplo, si se elimina una entidad, los esquemas externos que no se refieran a ella no se verán afectados.
- **Independencia física:** la capacidad de modificar el esquema interno sin tener que alterar ni el esquema conceptual, ni los externos. Por ejemplo, se pueden reorganizar los archivos físicos con el fin de mejorar el rendimiento de las operaciones de consulta o de actualización, o se pueden añadir nuevos archivos de datos porque los que había se han llenado. La independencia física es más fácil de conseguir que la lógica, pues se refiere a la separación entre las aplicaciones y las estructuras físicas de almacenamiento.



## 4 Funciones de un SGBD

La función principal: Permitir a los usuarios las 4 operaciones fundamentales:

- Creación e inserción,
- Consulta
- Actualización
- Borrado

Con 2 objetivos:

1. proporcionándoles una visión abstracta de los datos (esconder como se almacenan y mantienen los datos)
2. Velocidad (buen tiempo de respuesta), seguridad (que sea verdad lo que resulta) y. Consistencia.



## 5 Objetivos y características de un SGBD

Para conseguir dichas funciones:

- **Independencia física y lógica.** Estructuras físicas independientes de la lógica y viceversa, acceso a datos sin necesidad de conocer la estructura interna.
- **Un catálogo o diccionario:** Datos sobre los datos.
- **Mantener la integridad y consistencia** Garantizar que la transacción se haga o se rechace.
- **Acceso compartido a la BD,** controlando la interacción entre usuarios concurrentes.
- **Mecanismos de respaldo y recuperación** para restablecer la información en caso de fallos en el sistema
- **Cumplir restricciones.** Reglas de la realidad
- **Administración centralizada. Herramientas de administración.**
- **Acceso controlado a los datos de la BD** mediante mecanismos de seguridad de acceso a los usuarios.

## 6 Componentes de un SGBD

Son los elementos que deben proporcionar los servicios comentados en la sección anterior. No se puede generalizar ya que varían mucho según la tecnología. Sin embargo, es muy útil conocer sus componentes y cómo se relacionan cuando se trata de comprender lo que es un sistema de bases de datos.

El SGBD es la aplicación que interacciona con los usuarios de los programas de aplicación y la base de datos. En general, un SGBD suele incluir los siguientes componentes:

- **Lenguaje de definición de datos (DDL: Data Definition Language)**  
Sencillo lenguaje artificial para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones
- **Lenguaje de control de datos (DCL: Data Control Language)**  
Encargado del control y seguridad de los datos (privilegios y modos de acceso, etc.). Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos. Todo esto se almacenará en la base de datos.
- **Lenguaje de manipulación de datos (DML: Data Manipulation Language)**  
Para la inserción, actualización, eliminación y consulta de datos. Para tal fin el lenguaje por excelencia es el conocido SQL (Structured Query Language). Incluye instrucciones para los tres tipos de lenguajes comentados y por su sencillez y potencia se ha convertido en el lenguaje estándar de los SGBD relacionales.
- **Diccionario de datos**  
Esquemas que describen el contenido del SGBD incluyendo los distintos objetos con sus propiedades.



- Objetos: tablas base y vistas (tablas derivadas)
  - Consultas.
  - Dominios y tipos definidos de datos.
  - Restricciones de tabla y dominio y aserciones.
  - Funciones y procedimientos almacenados.
  - Disparadores– o triggers.
- distintas herramientas para:
  - Seguridad: de modo que los usuarios no autorizados no puedan acceder a la base de datos.
  - Integridad: que mantiene la integridad y la consistencia de los datos.
  - El control de concurrencia: que permite el acceso compartido a la base de datos.
  - El control de recuperación: que restablece la base de datos después de que se produzca un fallo del hardware o del software.
  - Gestión del diccionario de datos (o catálogo).
  - Importación/exportación de datos (migraciones).
  - Replicación (arquitectura maestro-esclavo).
  - Sincronización (de equipos replicados).
- Optimizador de consultas
 

Para determinar la estrategia óptima para la ejecución de las consultas.
- Gestión de transacciones
 

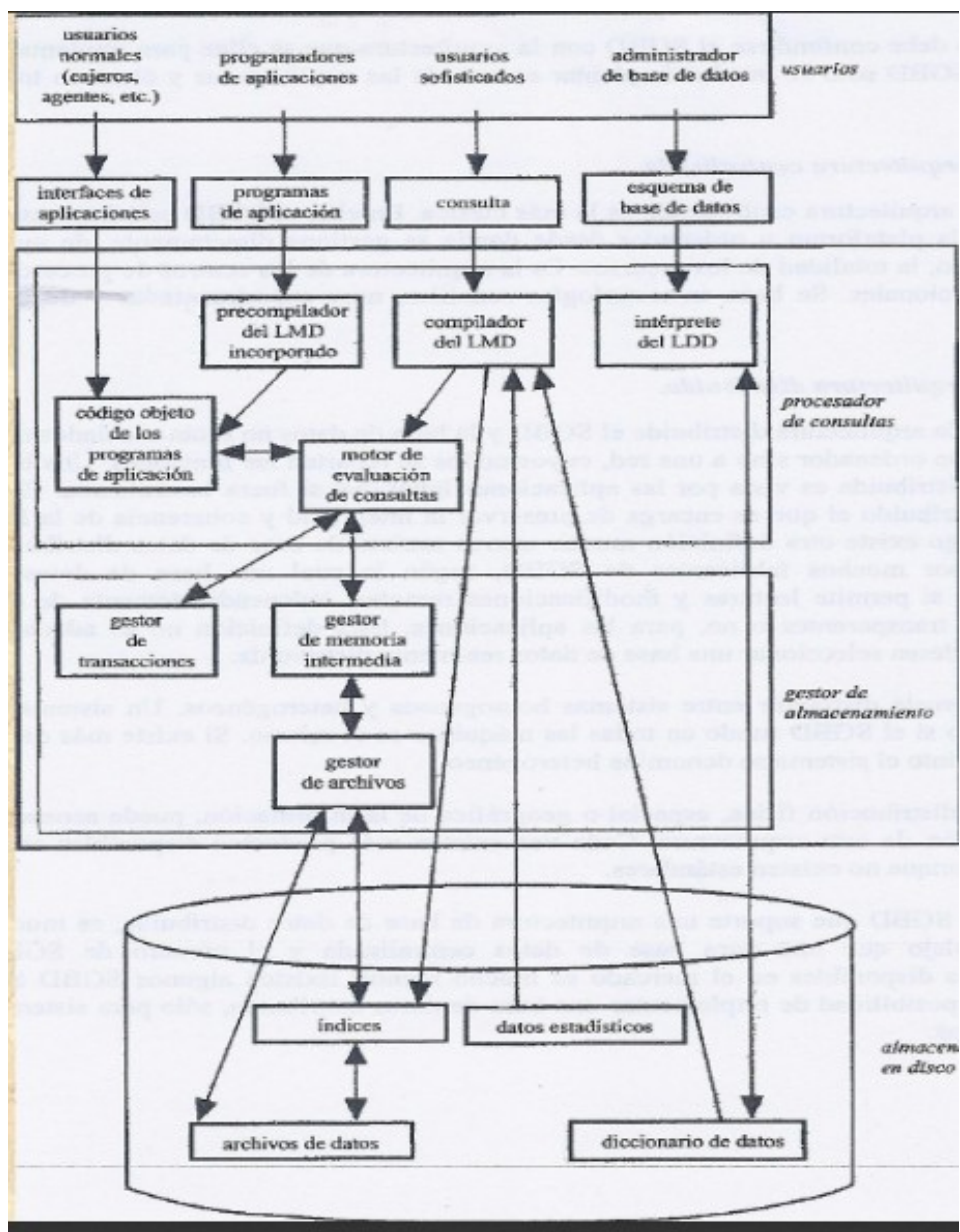
Este módulo realiza el procesamiento de las transacciones.
- Planificador (scheduler)
 

Para programar y automatizar la realización de ciertas operaciones y procesos.
- Copias de seguridad
 

Para garantizar que la base de datos se puede devolver a un estado consistente en caso de que se produzca algún fallo.







## 7 Usuarios

- Administrador: Diseño físico
- Analistas – Diseñadores: Diseño lógico
- Programadores: Implementa los programas
- Usuarios finales: Utilizan el resultado final.

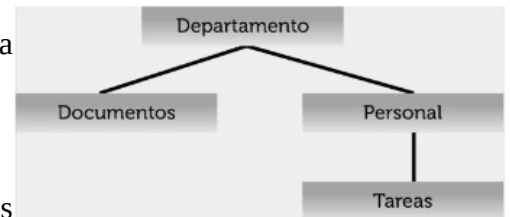


## 8 Tipos de SGBD

Según el modelo:

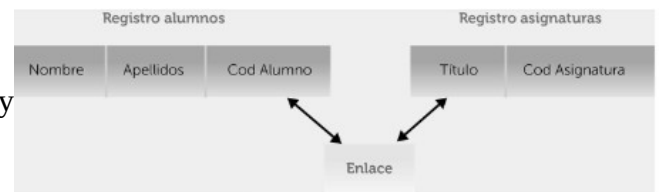
- Jerárquico:

- Son las más antiguas, relacionaba los datos usando una estructura en forma de árbol (relaciones padre/hijo)
- Usan segmentos y arcos
- Totalmente obsoleto por no poder reflejar relaciones complejas



- Red:

- Creado en los años 60, predominó hasta los 80
- La versión más popular es el modelo Codasyl
- Estructura y relaciona los datos usando nodos y enlaces
- Capaz de reflejar cualquier tipo de relación
- Manejo complejo y poco intuitivo



- Relacional:

- El más utilizado en la actualidad
- Usa tablas como estructura fundamental
- Su lenguaje de trabajo es el SQL
- Fácil de entender y manipular (comparado con los anteriores)
- Los SGBD de este tipo son muy potentes

- Orientadas a Objetos:

- Basados en la Programación Orientada a Objetos
- Su principal característica es que usa como estructura fundamental los objetos que permiten unir datos (atributos) y operaciones (métodos)
- Fáciles de asociar a lenguajes modernos como Java, C# o C++
- No han tenido mucho éxito

- Bases de datos Objeto relacionales:

- Intenta unir lo mejor de los dos modelos anteriores
- Se basan en la ISO SQL 2000
- Bases de datos relacionales que utilizan características de las orientadas a objetos



- Casi todas las bases de datos relacionales actuales son de este tipo (Oracle, SQL Server, DB2, ...)
- **Bases de datos NoSQL:**
  - El modelo relacional no funciona ante demandas enormes
  - Para esta situación se usan las llamadas bases de datos NoSQL
  - No usan SQL, almacenan en formatos como XML y JSON

## 9 SGBD comerciales y libres

### 9.1 Tipos de licencias

- **GPL (GNU General Public License).** Es la licencia más utilizada en el mundo del software de código abierto. Fue creada por Richard Stallman para el proyecto GNU, por lo tanto es la licencia del sistema Linux. Es la Free Software Foundation la entidad encargada de actualizar y revisar esta licencia.

Esta licencia permite:

- Modificar y distribuir el software bajo esta licencia.
- Cobrar por la distribución del software.
- **Licencia BSD.** Es la que se creó para los sistemas BSD (Berkeley Software Distribution). Es menos restrictiva que la anterior. Permite modificar y distribuir el software bajo esta licencia en la forma que queramos siempre que se cumpla
  - Si distribuimos el software debemos mantener el aviso de copyright anterior del software en el que se basa nuestra distribución.
  - No podemos usar el nombre de los autores originales ni de sus colaboradores para apoyar o promocionar una distribución modificada del software original.

Hay que tener en cuenta que incluso podemos modificar software con licencia BSD y distribuirle bajo una licencia de software cerrado. El caso más típico de esta idea es el sistema Mac OS de Apple que se basa originalmente en un sistema BSD.

- **Licencia MIT.** Licencia creada por el Instituto Tecnológico de Massachusetts, para sus distribuciones de software, concretamente se utilizó para el sistema X Window System. Es, como la anterior, muy permisiva. Permite modificar y redistribuir el software en la forma que el usuario desee, basta con incluir el aviso de copyright de esta licencia (que incluso se puede modificar).
- **Licencia Apache.** Presente en todo el software distribuido por la fundación Apache. Solo exige que se avise al usuario que el software original utilizaba licencia Apache, pero se permite que una distribución de software sea más restrictiva que el original (al estilo de las dos licencias anteriores).



- **Dominio público.** Es la licencia más permisiva de todas. De hecho un software bajo esta denominación no tiene ningún derecho de autor o restricción de uso. Es la licencia de uso de las obras (sean del tipo que sean) a las que le caducan los derechos de autor.

## 9.2 SGBD de código abierto

- **MySQL.** Inicialmente creada por la empresa MySQL AB, posteriormente comprada por Sun Microsystems que, a su vez, fue comprada por Oracle. Ha sido considerada como la principal SGBD de la comunidad de programadores de código abierto y de hecho en Internet sigue siendo la principal base de datos asociada a una aplicación web. Mantiene su licencia de tipo GPL, pero posee una segunda licencia cerrada para opciones de compra comercial con soporte.

Es muy popular por su histórica asociación con el lenguaje PHP, por su buena estabilidad, gran escalabilidad, e incluso uso de transacciones y lenguaje procedimental; además de ser un producto con infinitud de plataformas posibles para su instalación.

- **MariaDB.** Derivada la anterior por Michael Widenius, fundador de MySQL. La idea es disponer de un producto compatible con MySQL que mantenga la licencia GPL de forma perdurable. Esta siendo muy exitosa y está reemplazando a MySQL en numerosas instalaciones de Linux.
- **PostgreSQL.** Versión de código abierto basada en el producto Ingres de la Universidad de Berkeley. Usa licencia de tipo MIT. Es muy potente y, probablemente, sea el SGBD relacional más respetuoso con los estándares. Hoy en día, está considerada como la más potente de las bases de datos de código abierto y a partir de su núcleo se han creado otros productos de gestión de bases de datos.
- **SQLite.** Creada por el desarrollador Richard Hipp, usa una licencia de dominio público. Presume de ser totalmente relacional, compatible con ACID (gestión de transacciones completo) e incluso incluir un lenguaje procedimental (al estilo de PL/SQL). Todo ello en apenas 300KB de espacio.

Está escrita en C y guarda todo lo que necesita en un solo fichero para cada base de datos. Su ligereza y potencia aceptable la han hecho muy popular en muchas aplicaciones e incluso en los sistemas operativos móviles (casi todos la integran).

- MongoDB (NoSQL)
- Apache Cassandra
- Hbase

## 9.3 SGBD de código cerrado

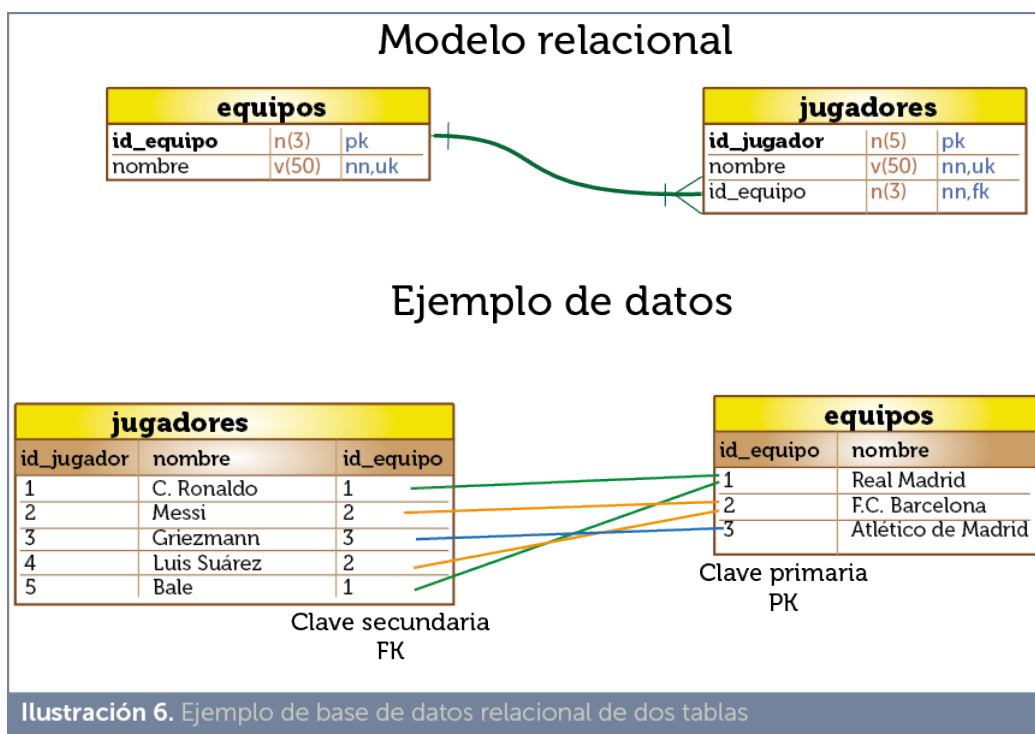
- Oracle
- DB2
- SQLServer



## 10 Repaso del modelo relacional

El Modelo Relacional fue enunciado por Edgar F. Codd en los años 70 y, todavía, sigue siendo el modelo más utilizado por los Sistemas Gestores de Bases de Datos comerciales. Los detalles fundamentales de este modelo son:

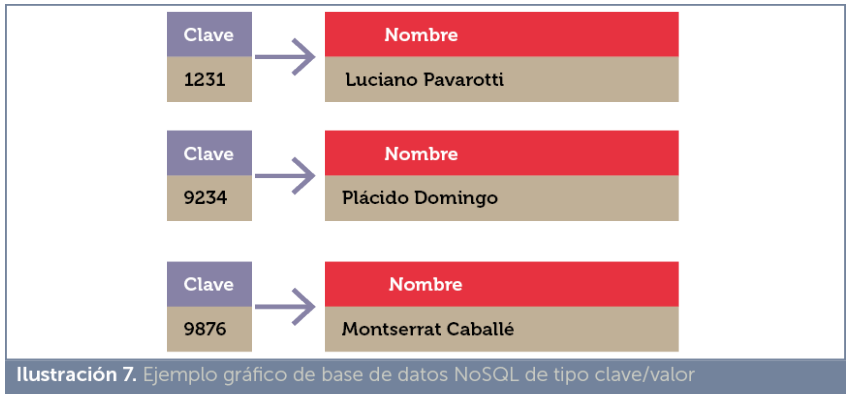
- Los datos se organizan en tablas. Cada tabla contiene datos referidos a un elemento distinguible en el mundo real (personas, facturas, contratos, etc.)
- Las tablas están formadas por filas y columnas. Las columnas indican un atributo (nombre, primer apellido, salario, etc.), cada fila es un ejemplar de cada elemento del mundo real. Por ejemplo, si una tabla se usa para almacenar los datos de los trabajadores, cada fila representa los datos de un trabajador.
- Una columna, o un conjunto de columnas, se establecen como clave principal o primaria de la tabla, de modo que los datos que contienen son distintos en cada fila de la tabla.
- Para poder relacionar datos de diferentes tablas, se usan claves secundarias o foráneas. Datos que se refieren a claves primarias de otras tablas.
- En las tablas relacionales se pueden aplicar estas restricciones:
  - Clave primaria (Primary Key, PK). Las columnas marcadas así, identifican cada fila y además no pueden quedar vacías ni repetir valores en diferentes filas.
  - Unicidad (Unique, UK). Las columnas marcadas con ella no pueden repetir valores en diferentes filas.
  - Obligatoriedad (Not Null, NN). Las columnas con esta restricción, obligatoriamente deben de contener algún valor (no pueden quedar vacías).
  - Integridad referencial (Foreign Key, FK). Los valores de esas columnas deben de corresponderse con los de las columnas de la clave principal con cuya tabla se relacionan.



## 11 Bases de datos NoSQL

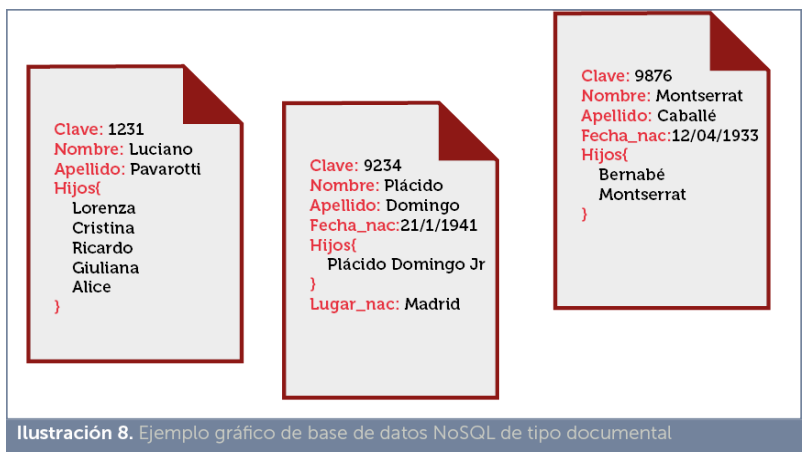
Se consideran dentro de esta clasificación a estos tipos de bases de datos:

- **Clave/valor.** Basadas en el funcionamiento de los arrays asociativos. Se asocia una clave (no repetible) a uno o varios valores.



- Amazon Dynamo DB. Desarrollada por la empresa Amazon, es un proyecto propiedad de esta empresa implementada para gestionar la enorme cantidad de transacciones y como una de las bases de su negocio en la nube. Almacena los datos en estructuras clave, valor: los valores son elementos binarios y, por lo tanto, opacos salvo desde la propia base de datos.
- Redis. Posee licencia BSD. Usa un modelo de asociación de claves a un contenido indexado. Usa la RAM de forma intensiva para almacenar los datos. El resultado de su arquitectura es una base de datos muy veloz. Por ello cada vez es más utilizada
- Oracle NoSQL. SGBD de Oracle que permite modelado de datos de forma relacional, como documentos JSON o como datos de tipo clave/valor.

- **Almacenes documentales.** En ella, la información importante utiliza un formato documental; es decir, un formato que encapsule la información y sus metadatos. Los datos se almacenan usando XML, JSON o incluso formatos binarios como PDF o Microsoft Office.



Los documentos se asocian a un valor clave (key) que permite su indexación. Algunas bases de datos comerciales de tipo documental son:

- MongoDB. Todavía es el SGBD de tipo NoSQL más utilizado. Tiene licencia GNU y utiliza un formato documental llamado BSON que es una versión binaria del lenguaje JSON. Las funcionalidades de acceso y gestión de datos y metadatos se realizan mediante JavaScript.
- Apache CouchDB. Dentro de la familia Apache (y usando licencia de código abierto Apache) es una base de datos NoSQL orientada a almacenar documentos. Usa formato JSON para



almacenar los datos y las consultas se realizan mediante JavaScript. Permite gestionar la información mediante **peticiones HTTP de tipo REST.**

- **Almacenes de columna ancha (wide column stores).** Se trata, en realidad, de una modalidad de base de datos de tipo clave/valor en la que los datos se almacenan en tablas, filas y columnas. La diferencia con las relacionales es que el nombre y tipo de las columnas varían de fila a fila.

Fila	Clave	Datos personales		Otros	
1	1231	Nombre	Luciano	País	Italia
		Apellidos	Pavarotti		
2	9234	Nombre	Plácido	Ciudad	Madrid
		Apellidos	Domingo	País	España
		Fecha_n	21/1/1941		
3	9234	Nombre	Montserrat	Ciudad	Barcelona
		Apellidos	Caballé	email	montse@gmail.com
		Fecha_n	12/4/1933		

Ilustración 9. Ejemplo gráfico de base de datos NoSQL de tipo clave/valor

- Google BigTable. Base de datos propietaria utilizada para muchos de los servicios de almacenamiento de Google. Los datos se almacenan en una estructura multidimensional de tres claves (fila, columna y fecha) y permiten ser particionados. El modelo físico se basa en el Google File System, modelo de archivos propietario de Google.
- Apache Cassandra. Con licencia Apache, es muy popular y de contrastada potencia. Se desarrolló inicialmente por facebook. Los datos se almacenan en tuplas sin relaciones de integridad. Presume, además, de un crecimiento exponencial en estos últimos tiempos y de ser el motor de base de datos de servicios como twitter o Netflix.
- Apache HBase. Licencia Apache. Se basó en BigTable. Se utiliza en numerosas aplicaciones basadas en datos. HBase es parte del proyecto Hadoop de Apache para el proceso de Big Data.



## 12 Actividades

### 12.1

1. Averigua y explica el significado del término ACID en el contexto de los sistemas gestores de bases de datos.
2. Busca al menos tres diferencias importantes entre SQL Server y MySQL.
3. Averigua en qué consiste y para qué sirve la minería de datos
4. Busca, resume y comenta opiniones en distintos foros o páginas sobre los sistemas MySQL, SQL Server y Oracle.
5. ¿Qué limitaciones tiene la versión Oracle Express Edition respecto a la extendida?
6. ¿Qué lenguaje específico usa Oracle para implementar el lenguaje SQL
7. Averigua el significado de los siguientes acrónimos en el contexto de los SGBD (SAP, ERP y DSS, LMD, LDD).
8. Haz un listado de al menos tres sistemas gestores de bases de datos libres (open source) y tres comerciales indicando tres de sus características principales. Discute los motivos por los que consideras que algunas empresas ofrecen productos de software gratuitos.

### 12.2

1. Comenta qué se entiende por software libre considerando aspectos como:
  - Gratuidad.
  - Código fuente.
  - Uso comercial
2. Lista al menos 3 ventajas e inconvenientes de los productos de pago respecto a los libres.
3. ¿Qué tiene que ver la administración SGBD con el diseño de bases de datos?
4. Cita al menos 3 ventajas de usar bases de datos frente a los tradicionales sistemas de ficheros.
5. Enumera al menos tres objetos típicos de una base de datos indicando su función.
6. ¿Qué es una base de datos distribuida?
7. Indica resumidamente las fases involucradas en el desarrollo de una base de datos desde su concepción hasta su puesta en marcha.
8. ¿Para qué sirve un disparador en un SGBD?
9. Explica con tus palabras qué es el diccionario de datos en un SGBD
10. Eres administrador de la base de datos. Indica un problema y su posible solución que te pueda surgir considerando dos casos: una base de datos con miles de usuarios y centrada en





consultas, como un buscador, y otra de venta online con miles de usuarios y operaciones por segundo

