

# INTRODUCCION

---

## Bases de Datos Relacionales

# Indice de Contenidos

---

- ☐ Concepto de “Motores de DB Relacionales”.
- ☐ Terminología utilizada en DB Relacionales.
- ☐ Componentes de una instancia.
- ☐ Archivos físicos y lógicos.
- ☐ Metadatos o diccionario de datos.

# Objetivos de la Clase

---

- ☐ Bases de Datos Relacionales y Gestor de Bases de Datos Relacionales.
- ☐ Instancia en un RDBMS.
- ☐ Componentes de una instancia.
- ☐ Archivos físicos y lógicos en una base de datos.
- ☐ Componentes de un RDBMS SQL Server y Oracle.
- ☐ Base de datos del sistema en Sql Server.
- ☐ Metadatos.
- ☐ Principales objetos de base de datos

# Concepto de Base de Datos Relacional

---

- ❑ Una Base de Datos, es una colección de datos, tablas y otros objetos. Los objetos de la Base de Datos ayudan a estructurar los datos y definir mecanismos para la integridad de los mismos .
- ❑ Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), Recopilación de objetos o relaciones.
- ❑ Cumple con el modelo relacional.

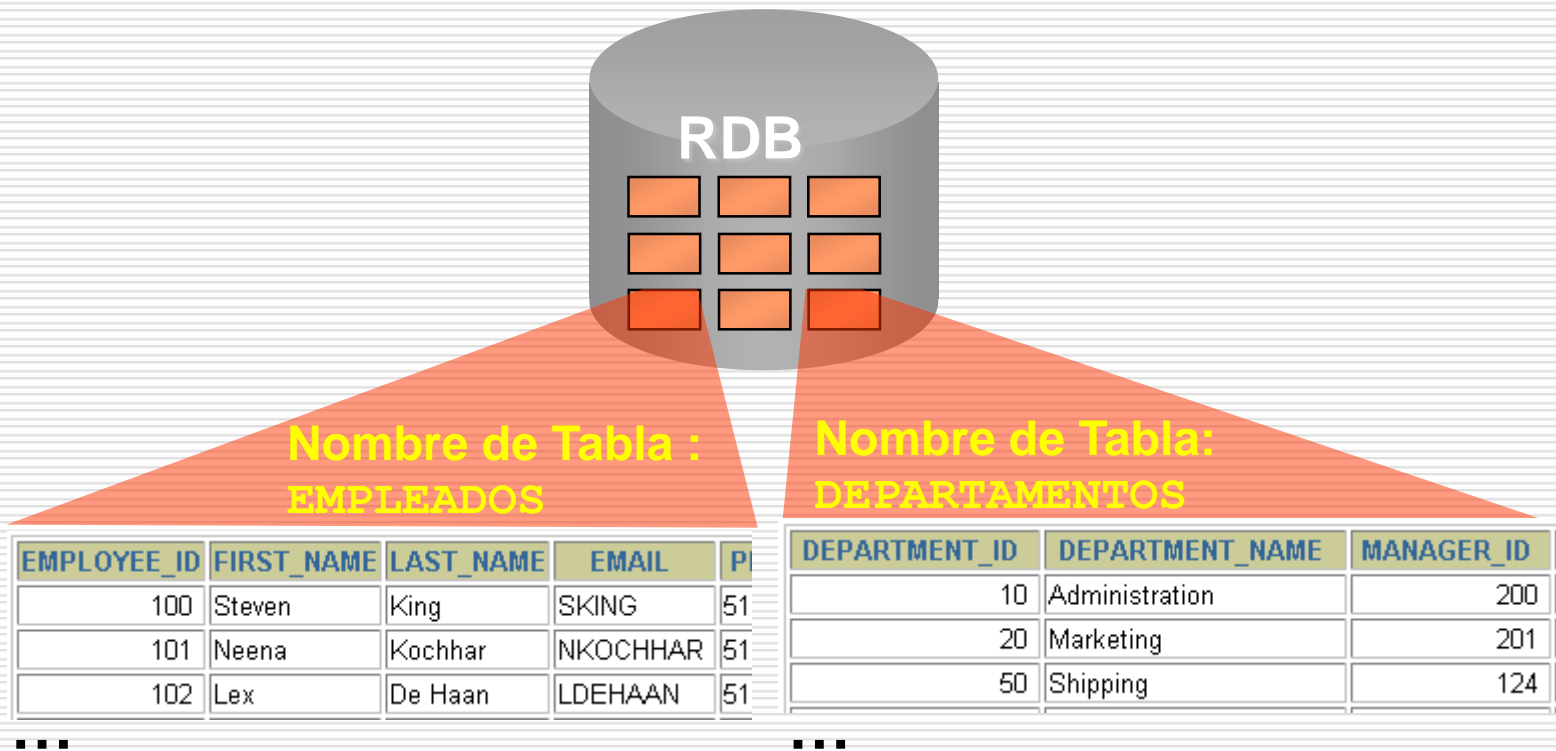
# Concepto de Base de Datos Relacional

---

- ❑ Una base de datos relacional se compone de varias tablas o relaciones.
- ❑ No pueden existir dos tablas con el mismo nombre.
- ❑ Cada tabla es a su vez un conjunto de registros (filas y columnas).
- ❑ La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- ❑ Las claves primarias son la clave principal de un registro dentro de una tabla y permiten la relación con otras tablas por medio de las claves externas.

# Definición de una Base de Datos Relacional

Una base de datos relacional es una recopilación de relaciones o tablas de dos dimensiones.



# Terminología en BD Relacionales

2

Columnas

3

Tipos de datos

4

Claves Externas

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	King	Steven	24000		90
101	Kochhar	Neena	17000		90
102	De Haan	Lex	17000		90
103	Hunold	Alexander	9000		60
104	Ernst	Bruce	6000		60
107	Lorentz	Diana	4200		60
124	Mourgos	Kevin	5800		50
141	Rajs	Trenna	3500		50
142	Davies	Curtis	3100		50
143	Matos	Randall	2600		50
144	Vargas	Peter	2500		50
149	Zlotkey	Eleni	10500	.2	80
174	Abel	Ellen	11000	.3	80
176	Taylor	Jonathon	8600	.2	80
178	Grant	Kimberely	7000	.15	
200	Whalen	Jennifer	4400		10
201	Hartstein	Michael	13000		20
202	Fay	Pat	6000		20
205	Higgins	Shelley	12000		110
206	Gietz	William	8300		110

5

Valores Nulos

6

1

Filas

# Relación de Varias Tablas


- Cada fila de datos de una tabla se identifica de forma única con una clave primaria (PK).
- Se pueden relacionar de forma lógica datos de varias tablas utilizando claves ajenas (FK).

Nombre de Tabla: **EMPLOYEES**

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
174	Ellen	Abel	80
142	Curtis	Davies	50
102	Lex	De Haan	90
104	Bruce	Ernst	60
202	Pat	Fay	20
206	William	Gietz	110

...

 Clave primaria

 Clave ajena

Nombre de Tabla: **DEPARTMENTS**

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

 Clave primaria



# Gestor de Base de Datos Relacional

---

- ❑ Del inglés: Relational database management system, abreviado (RDBMS).
- ❑ Son un tipo de software muy específico, dedicado a servir de interfaz entre la bases de datos relacionales, el usuario y las aplicaciones que la utilizan.

# Instancia en un RDBMS

---

- ❑ Es un servicio de aplicación autocontenida que implica archivos del sistema operativo, estructuras de memoria, procesos de segundo plano e información de registro.
- ❑ Una instancia esta representada por uno o mas procesos dentro del sistema operativo.
- ❑ En Windows puede ser un servicio y puede estar en ejecución o detenido dependiendo del gestor.

# Componentes de una instancia

---

- ❑ **Procesos Background:** Son procesos del sistema operativo que se utilizan para acceder a los objetos de la base de datos.  
Permiten compartir estos objetos con todos los usuarios.
- ❑ **Estructuras de memoria:** áreas especiales de memoria que permiten almacenar objetos de una base de datos y ayudan a mejorar el rendimiento.
- ❑ **Archivo de del sistema operativo:** permiten almacenar los datos y todas las modificaciones realizadas a una base de datos.

# INSTANCIA EN SQL SERVER

Instancia

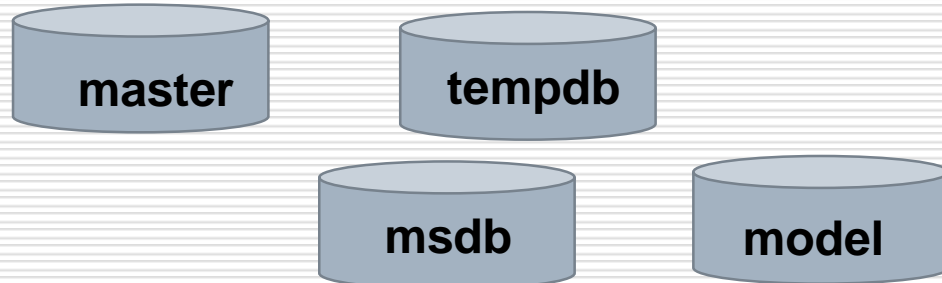
**es un**

Servicio de Windows

**Puede estar**

Apagado o Encendido

Cuando se instala SQL Server se crean 4 DB:



A nivel físico hay 2 tipos de archivos:



Data File: Archivo de Datos.

Transaction Log File: Cambios a los datos.

# INSTANCIA EN ORACLE

Instancia

**es un**

Servicio de Windows

**Puede estar**

Apagado o Encendido

Cuando se instala Oracle se crean 3 tablespaces:

System

SysAux

Temp

A nivel físico hay 3 tipos de archivos:

**Data File**

**Redo log file**

**Control File**

Data File: Archivo de Datos.

Redo Log File: Cambios a los datos.

Control File: Archivo de parámetros.

# INSTANCIA EN MySQL

Instancia

**Es un**



Proceso de Windows

**Puede ser**



Servicio de Windows

**Puede estar**



Apagado o Encendido

Cuando se instala MySQL se crean 4 BD:

mysql

performance\_schema

sys

information\_schema

A nivel físico (InnoDB):

Dir BD

ibdata1

ib\_logfile0

bin.0000001

ib\_logfile1

ibdata1: espacio de tablas compartido.

ib\_logfile: LOG transaccional

bin.00000: LOG Binario

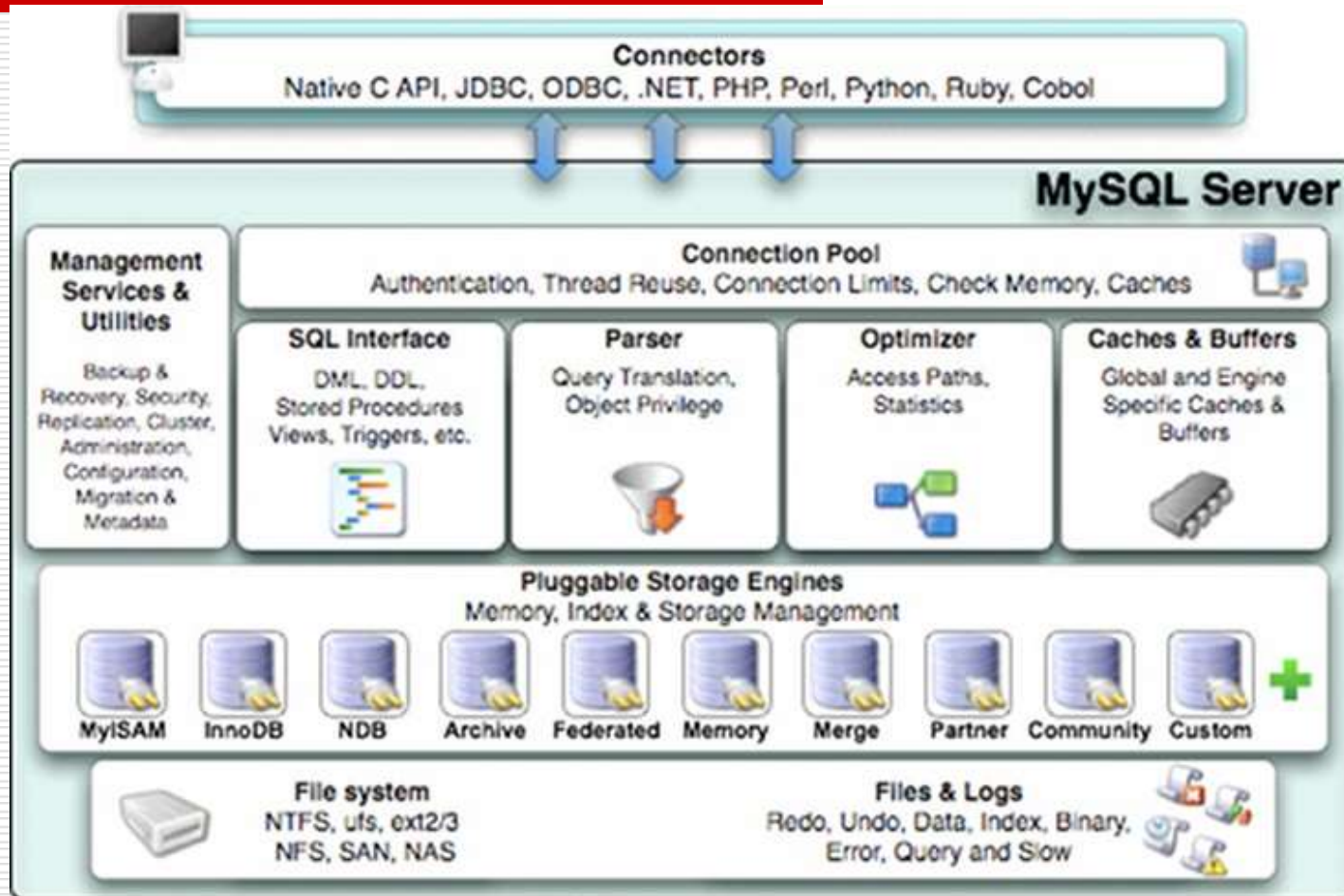
Directorio de base de datos

# BD del sistema **MySQL**

---

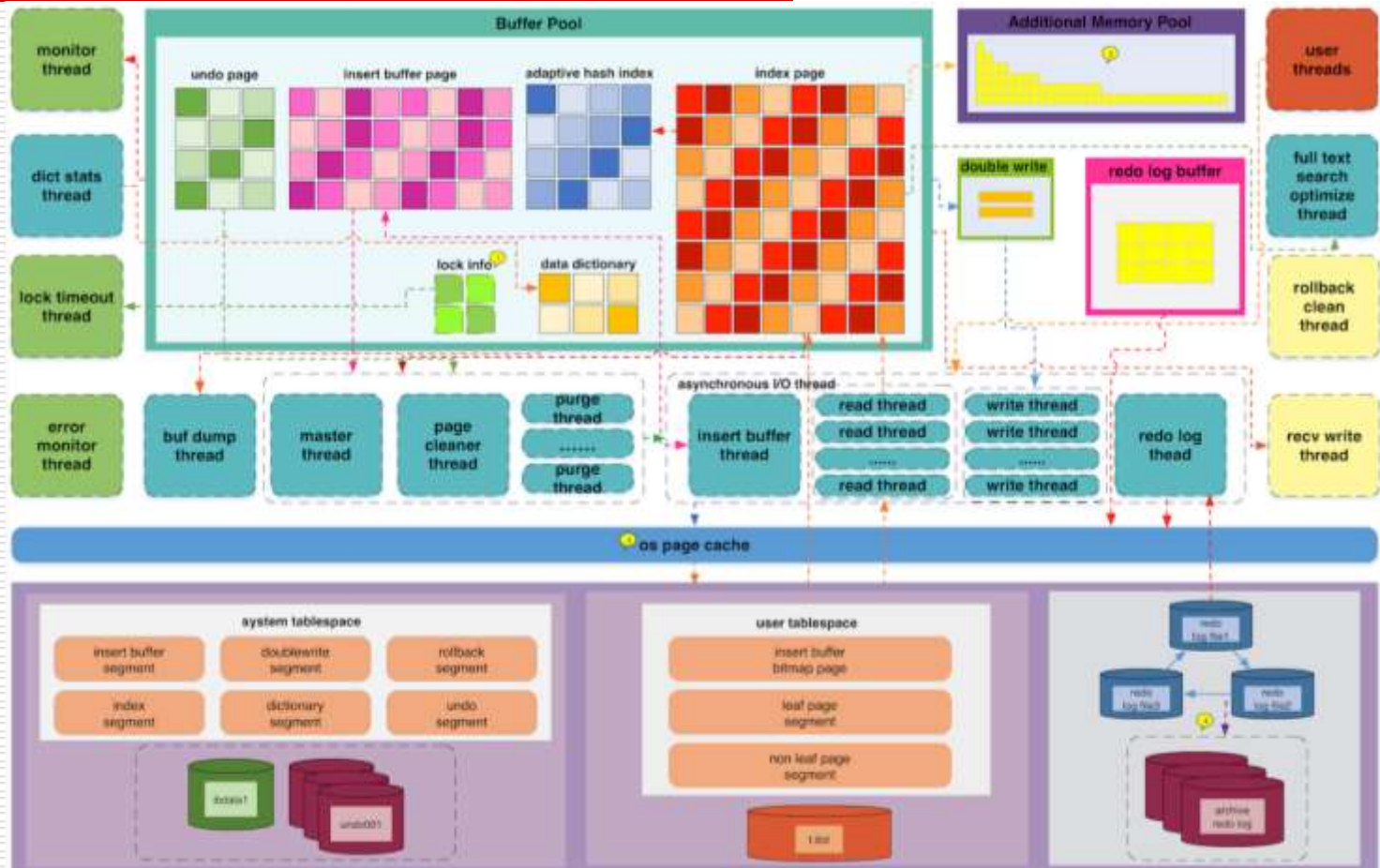
- ❑ **MYSQL:** describe los privilegios de acceso de los usuarios. El servidor almacena la información de privilegios en las tablas de permisos de esta base de datos, lee el contenido en memoria cuando arranca, y basa las decisiones de control de acceso en esta información (in-memory copies)
- ❑ **INFORMATION\_SCHEMA:** proporciona acceso a los **metadatos** de la base de datos. Contiene información sobre el servidor MySQL, tal como el nombre de una base de datos o tabla, el tipo de datos de una columna, o privilegios de acceso. Otros términos que se utilizan son **diccionario de datos** y el catálogo del sistema. Es una base de datos dentro de cada instancia de MySQL.
- ❑ **PERFORMANCE\_SCHEMA:** es un mecanismo para el seguimiento de la ejecución del servidor MySQL a bajo nivel, y da a los usuarios una visión de lo que sucede en el "background" cuando el servidor MYSQL está corriendo.
- ❑ **SYS\_SCHEMA** (ps\_helper): colección de vistas, procedimientos y funciones, diseñados para tener una lectura sencilla de los datos "crudos" del PERFORMANCE\_SCHEMA.

# Arquitectura MySQL





# Arquitectura MySQL InnoDB



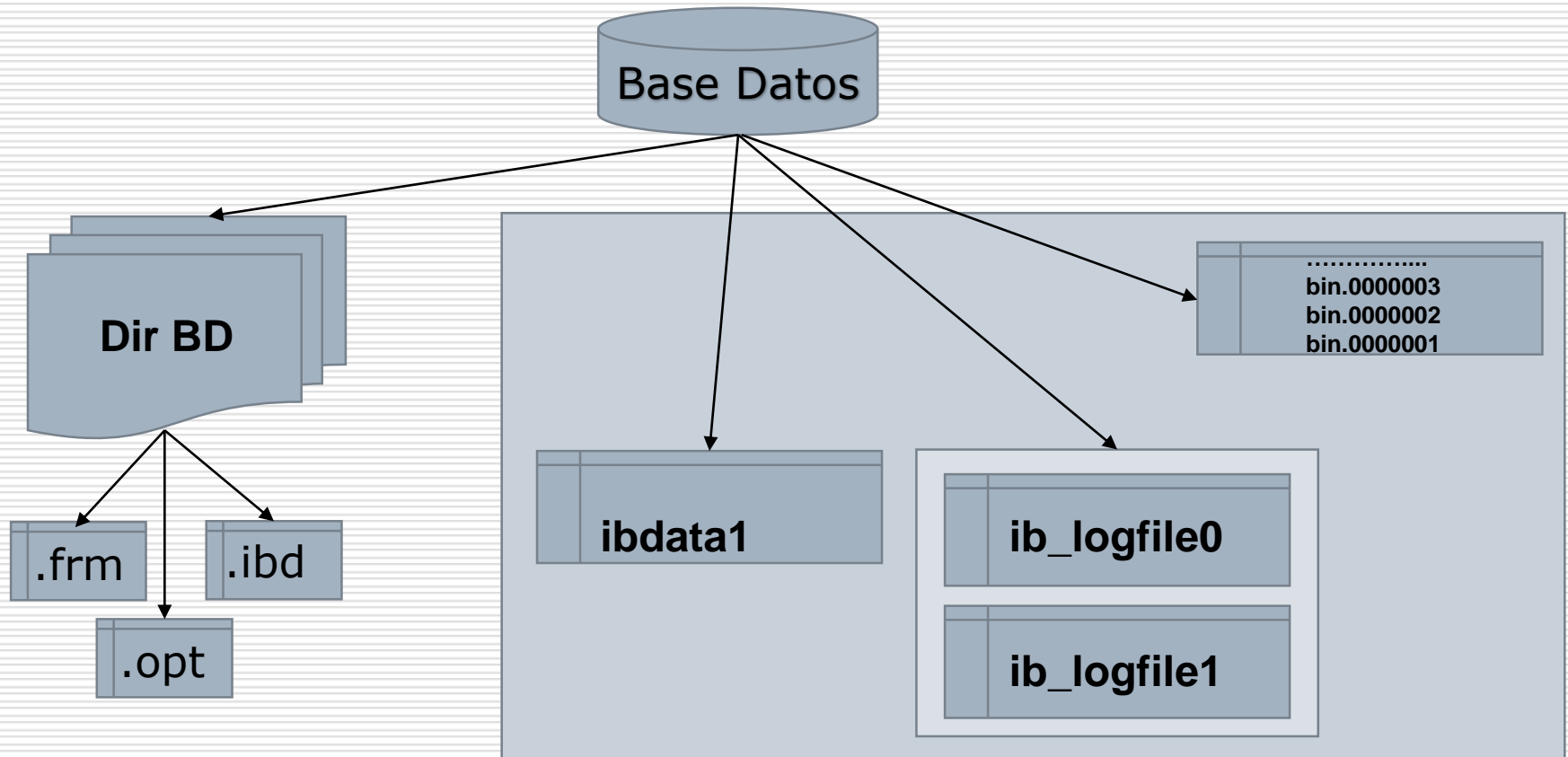
# Organización física

---

- ❑ Un archivo de base de datos no es mas que un archivo del sistema operativo. Una base de datos se distribuye en por lo menos dos archivos, aunque es muy probable que sean varios los archivos de base de datos que se especifican al crear o al modificar una base de datos.
- ❑ Algunas DBMS dividen su trabajo en archivos para datos y archivos para el registro de las transacciones (log).

# Organización física MySQL – InnoDB Storage Engine

---



# Organización física

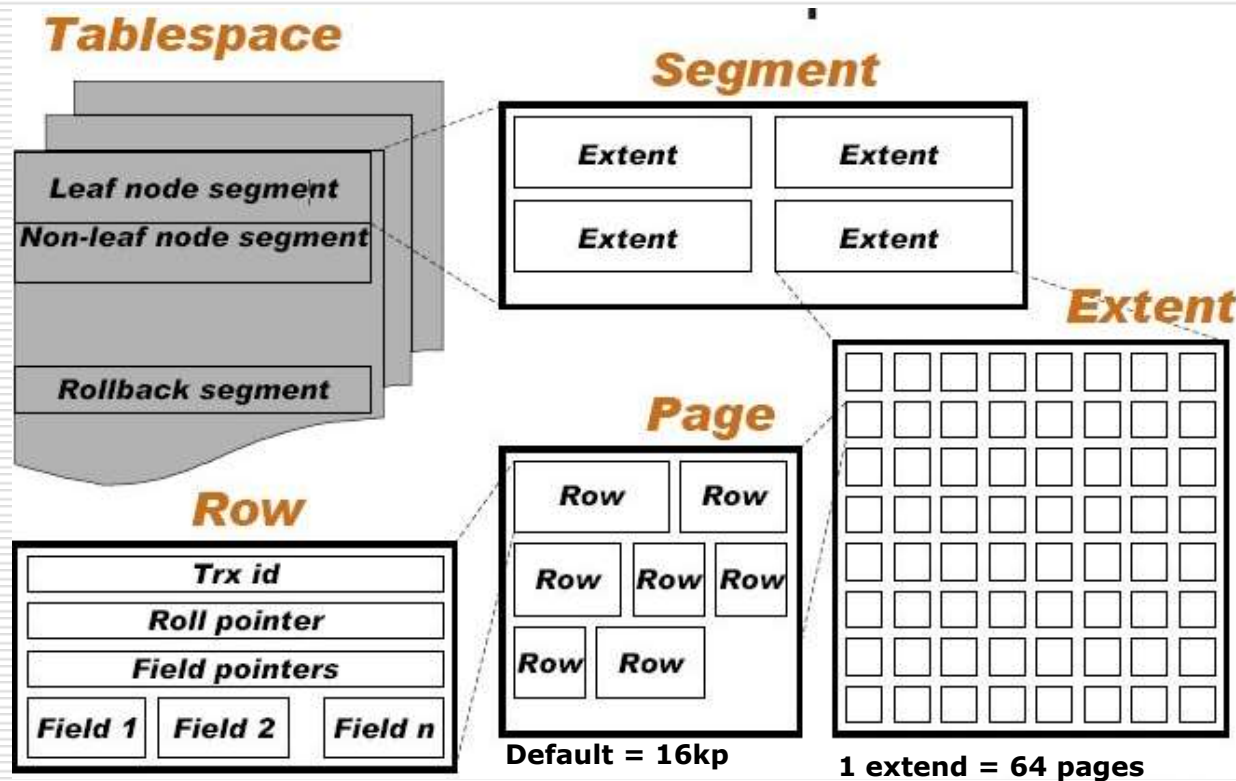
## MySQL – InnoDB Storage Engine

---

- ❑ **Ibadata1: Espacio de tablas compartido** que consiste en los ficheros ibdata, donde InnoDB coloca su diccionario de datos interno y los registros para deshacer cambios (undo logs). Si está habilitado, guarda el conjunto de datos de cada base de datos.
- ❑ **LOG de transacciones(redo log):** Una estructura de datos basada en disco utilizado para la recuperación ante una caída del motor de base de datos, permite corregir datos escritos por transacciones incompletas.
- ❑ **Directorio BD:**
  - **.ibd:** espacio de tabla generado para cada base de datos, si está habilitado.
  - **.opt** y **.frm** : Diccionario de datos, compatibilidad versiones anteriores.
- ❑ **LOG Binario:** sirve para guardar un registro de las consultas que afectan a los datos de la BD. Activando el log binario se puede reconstruir los datos de la base de datos en un 99% desde el último backup

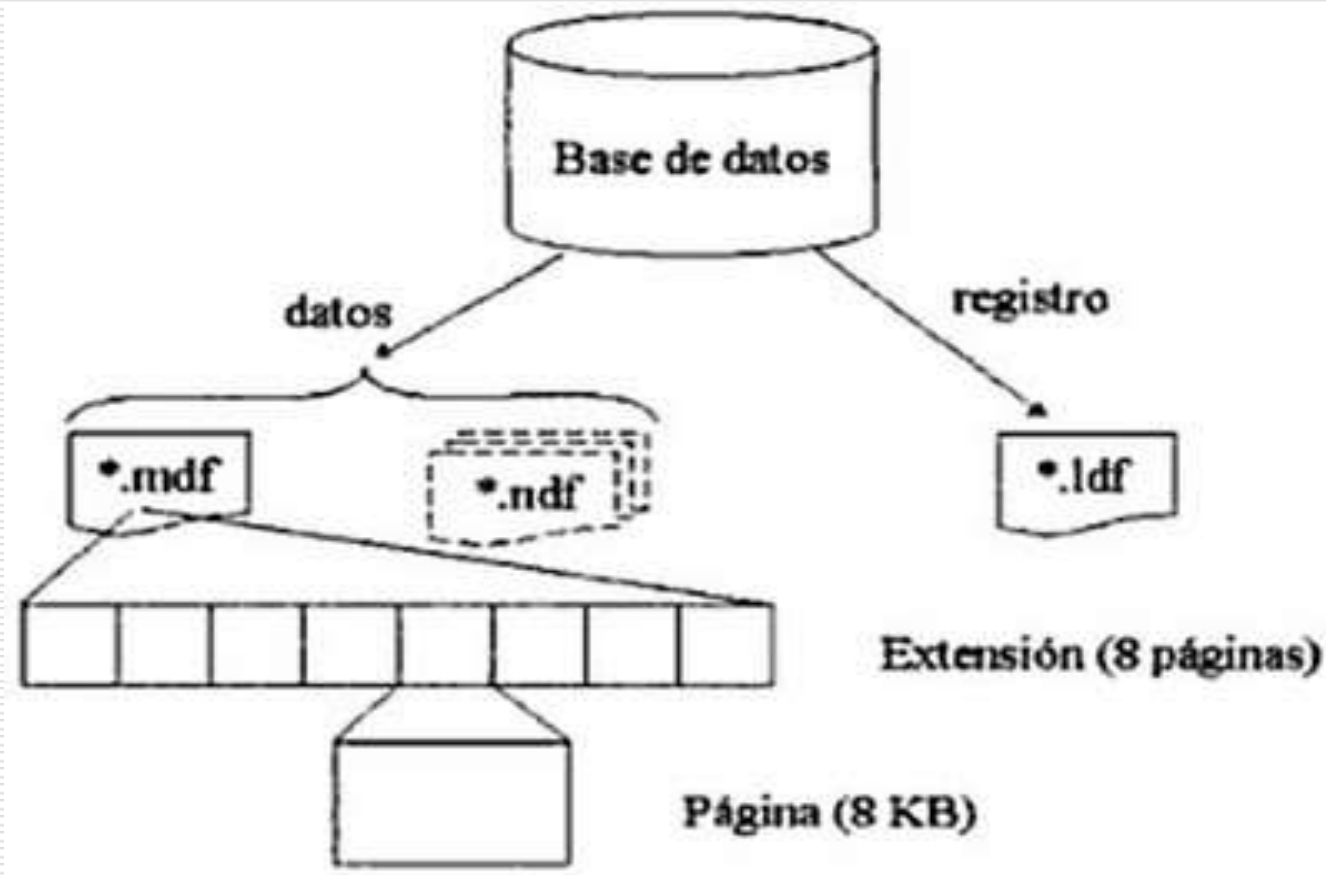
# Organización física

## MySQL – InnoDB Storage Engine



# Organización física **SQL SERVER**

---



# SQL SERVER

---

SQL Server permite los tres siguientes tipos de archivos:

- **Archivos de datos primarios:** Toda base de datos tiene un archivo de datos primario que realiza el seguimiento de todos los demás archivos, además de almacenar datos. Por convenio este archivo tiene la extensión MDF.
- **Archivos de datos secundarios:** Una base de datos puede tener cero o varios archivos de datos secundarios. Por convenio la extensión recomendada para los archivos de datos secundarios es NDF.
- **Archivos de registro (LOG):** Todas las bases de datos por lo menos tendrán un archivo de registro que contiene la información necesaria para recuperar todas las transacciones que suceden sobre la misma. Por convenio la extensión de este archivo es LDF.

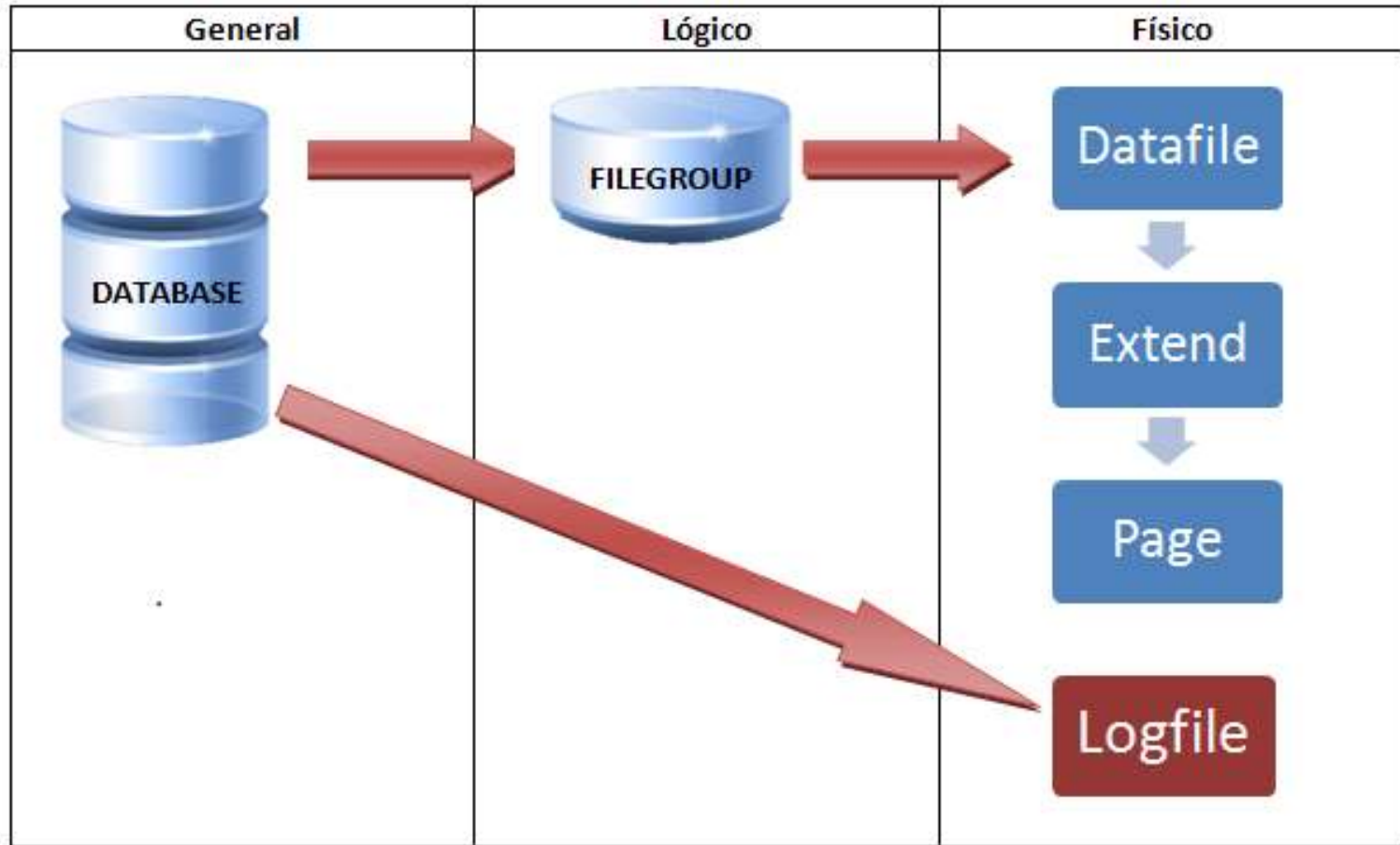
# Paginas y extensiones SQL SERVER

---

- La unidad básica de almacenamiento en SQL Server es la página o data page, el tamaño de cada page es de 8 KB, lo cual representa un total de 128 páginas por cada megabyte
- Las extensiones son la unidad básica de asignación de espacio a las tablas e índices. Constan de 8 páginas contiguas, es decir 64 KB. Lo cual representa 16 extensiones por MB.
- En SQL Server, las filas no pueden continuar en otras páginas.

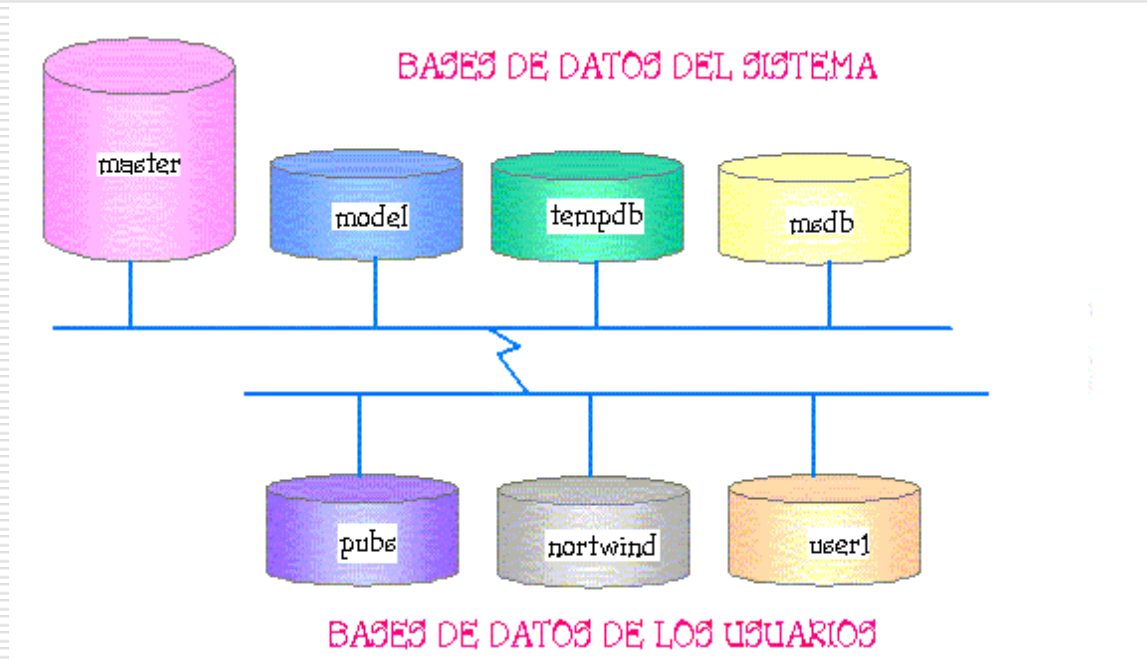


# Organización física del SQL SERVER



# DB del sistema SQL SERVER

---



# DB del sistema SQL SERVER

---

- ❑ **Master:** Bases de datos principal. Controla las bases de datos de usuario y la operación de SQL server de forma global al realizar el seguimiento de la información tal como las cuentas de usuario y los mensajes de error del sistema.
- ❑ **Model:** Bases de datos plantilla. Proporciona una plantilla o prototipo para las nuevas bases de datos de usuario. Cada vez que se crea una nueva base de datos se hace una copia de model como base de la nueva base de datos.
- ❑ **Msdb:** Base de datos utilizada por el servicio SQL Server Agent. Almacena información sobre copias de seguridad, mantenimiento y duplicación de bases de datos.
- ❑ **Tempdb:** Base de datos temporal. Es un espacio de trabajo. Se crea cada vez que inicia SQL Server. Utilizada para las tablas temporales creadas explícitamente, para las tablas que mantienen resultados intermedios y cursores estáticos.

# Archivo de LOG

---

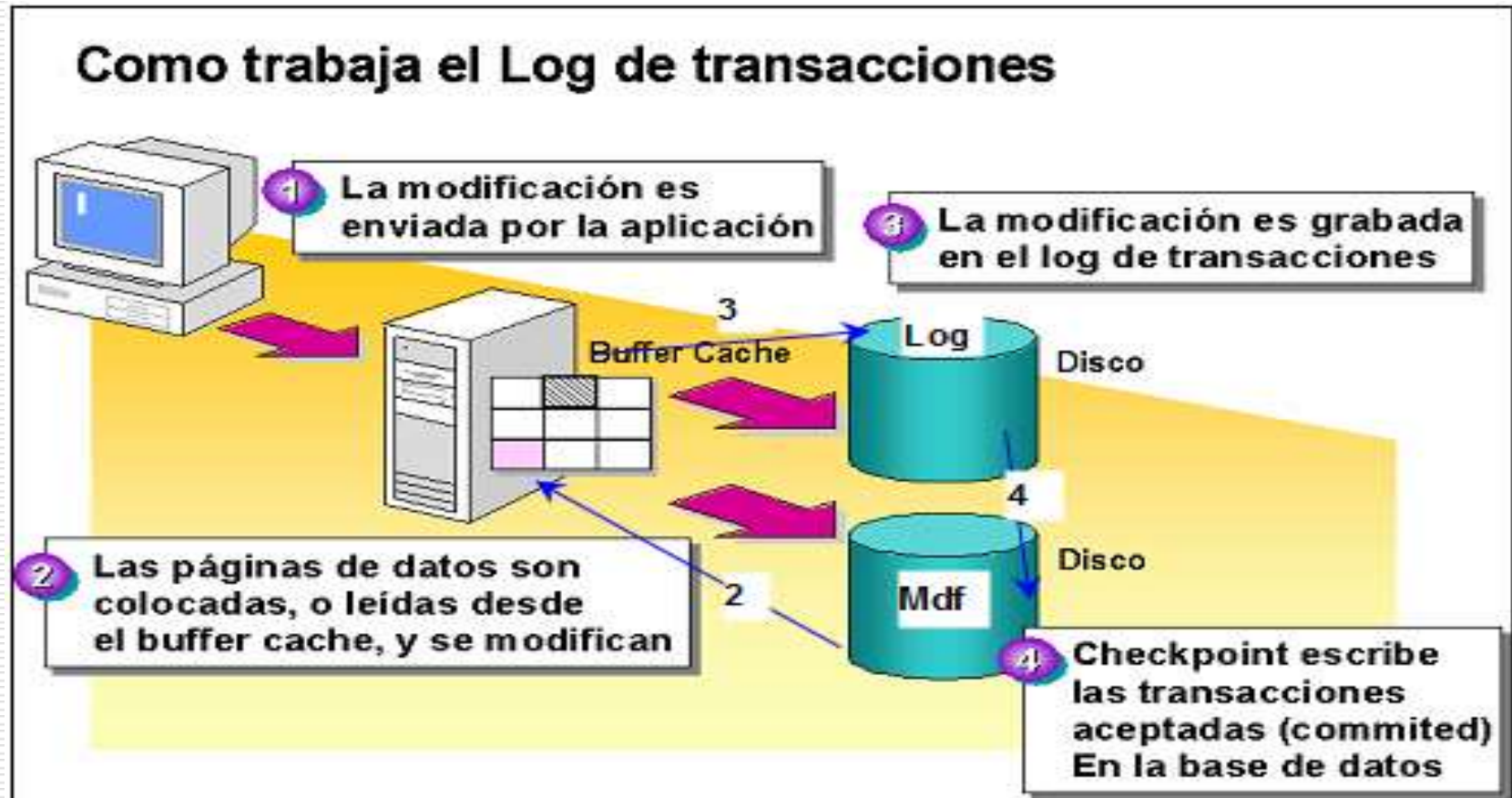
- El LOG de transacciones archiva todas las modificaciones de los datos tal cual son ejecutados. El proceso es como sigue:
- Una modificación de datos es enviada por la aplicación cliente.
- Cuando una modificación es ejecutada, las páginas afectadas son leídas del disco a memoria (Buffer Cache), si no están todavía por algún query previo.

# Archivo de LOG

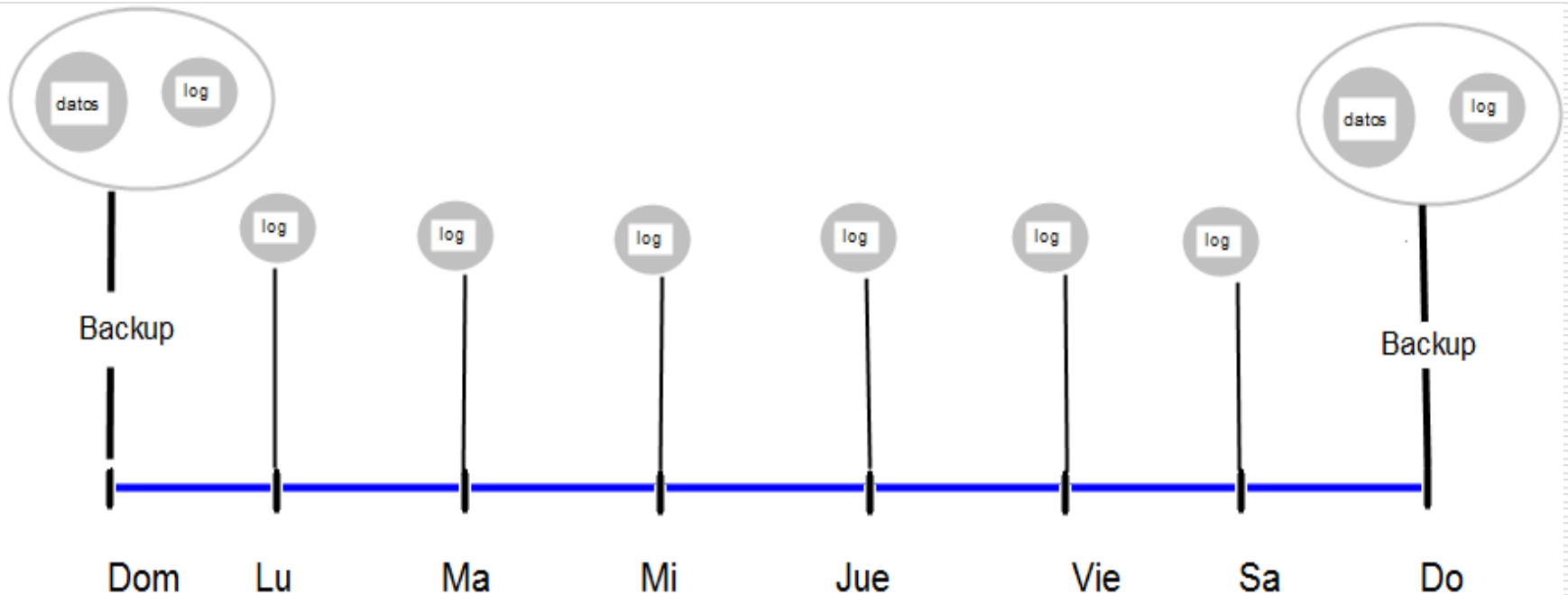
---

- Las modificaciones a los datos son archivadas primeramente en el LOG. El cambio siempre es escrito en el archivo de LOG antes que en el archivo de datos de la Base de Datos.
- El proceso "CHECKPOINT" (proceso Background) es el encargado de escribir todas las transacciones completadas en el archivo de LOG al archivo de datos de la Base de Datos en el disco.
- Si el sistema falla, automáticamente el proceso de recuperación usa el LOG de Transacciones para llevar hacia delante todas las transacciones comprometidas (COMMIT) y llevar hacia atrás alguna transacción incompleta (ROLLBACK).

# Archivo de transacciones LOG



# Archivo de transacciones (LOG)



# METADATOS

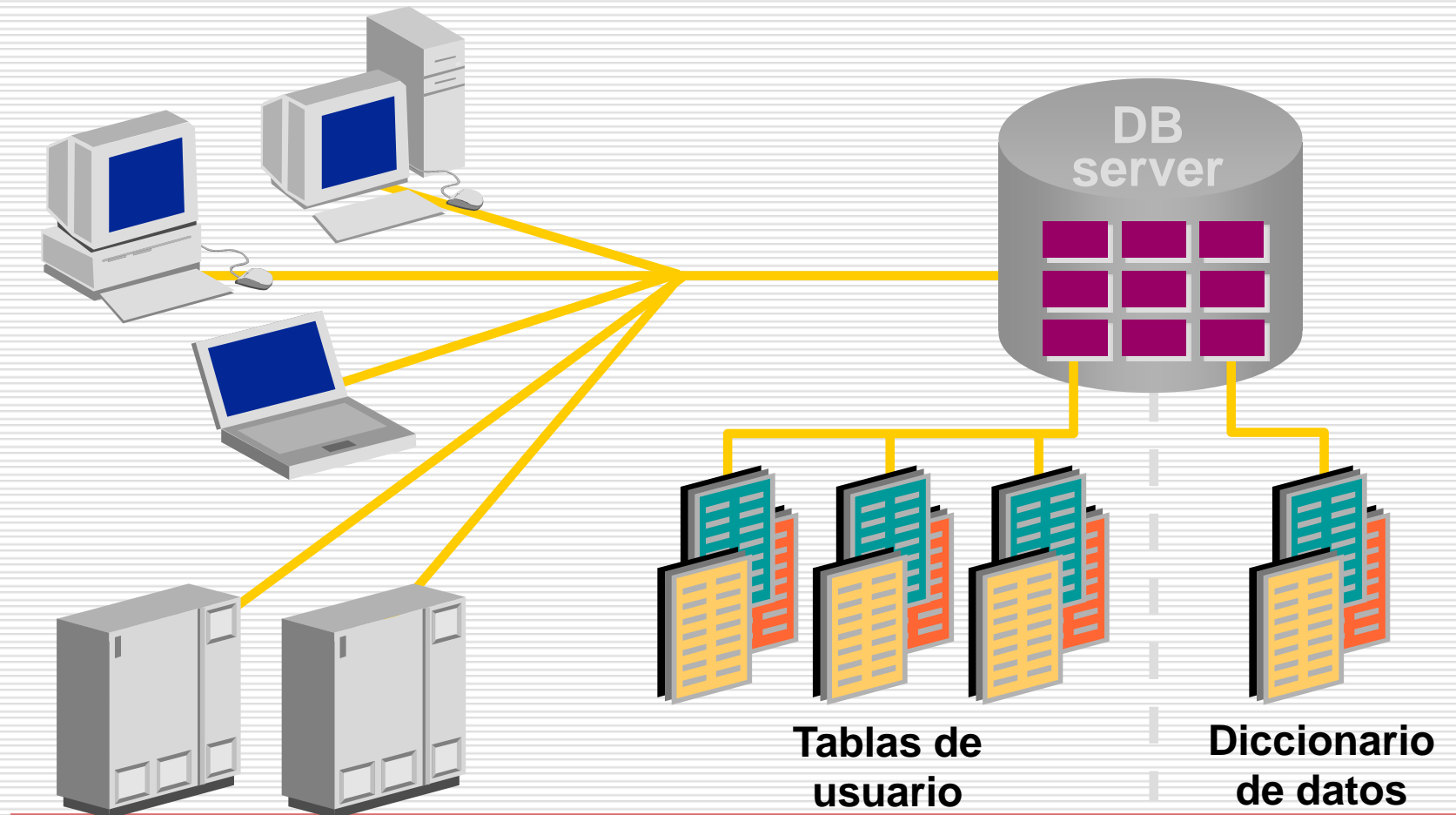
---

- ❑ Metadatos = diccionario de datos, se dice que es información sobre la información, por eso es un metadato.
- ❑ Son tablas, vistas y procedimientos almacenados que se definen en forma automática y permiten conocer entre otras cosas:
  1. Estructura lógica y física de la DB
  2. Los usuarios de la DB
  3. Restricciones de integridad sobre las tablas de la DB
  4. Espacio asociado a cada objeto en la DB
- ❑ Objetos: sys.database\_files ,Sys.databases, sys.tables, sp\_help, sp\_columns etc.

**EJ:** select a.total\_pages, a.used\_pages, object\_name(p.object\_id) objeto, p.rows  
from sys.allocation\_units a  
INNER JOIN sys.partitions p on a.container\_id = p.partition\_id  
INNER JOIN sys.objects AS o ON p.object\_id = o.object\_id  
where o.type = 'U'  
order by p.object\_id



# Gestor de DB Relacionales



# Objetos en una B. de Datos

---

Objeto	Descripción
Tabla	Unidad básica de almacenamiento; está formada por filas y columnas.
Vista	Representa lógicamente subconjuntos de datos de una o varias tablas.
Índice	Mejora el rendimiento de algunas consultas.
Sinónimo	Proporciona nombres alternativos a objetos.
.....	.....

# Comunicación con un RDBMS Utilizando SQL

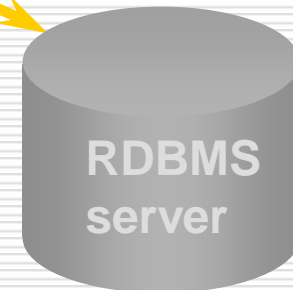
---

**Sentencia SQL  
ingresada.**

```
SELECT department_name  
FROM departments;
```

**Se envía la sentencia  
al RDBMS.**

DEPARTMENT_NAME
Administration
Marketing
Shipping
IT
Sales
Executive
Accounting
Contracting





# ESPACIO PARA PREGUNTAS