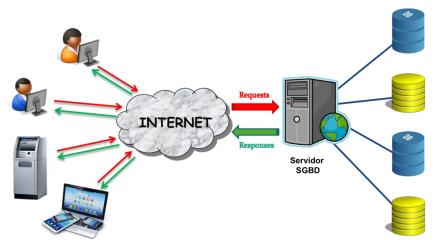


T1: ESTRUCTURA INTERNA

INTRODUCCIÓN

SGBD/DBMS (Database Management System)

sistema software que permite almacenamiento, modificación y extracción de info



Capa de Aplicación | Comunicación usuario - API

Capa Lógica | Funcionalidad Central SGBD

Capa Física | Almacenamiento físico datos uso fisiologías S.O.



COMPONENTES DE UN SGBD

- Lenguajes SGBD
- Diccionario de datos: info. sobre los datos q forman las BBDD
- Gestor de almacenamiento
- Procesador de consultas
- Seguridad e Integridad
- Adminin. de los BBDD (DBA - Data Base Administrator)

Lenguajes SGBD en SQL

ADMINS

- Lenguaje de Definición de Datos (DDL)
`CREATE, DROP, ALTER`

- Lenguaje de Control de Datos (DCL)
`GRANT, REVOKE, COMMIT, ROLLBACK`

USUARIOS

- Lenguaje de Manipulación de Datos (DML)
`INSERT, UPDATE, DELETE, SELECT`

Típos de usuarios

- Administradores → gestionar BBDD, control acceso, ...
- Diseñadores → diseño lógico
- Programadores
- usuarios finales → nivel de vistas o portes de lo BBDD

Servidor de BBDD

Fases del admin:

Gestión del software	<ul style="list-style-type: none"> • instalación del software • actualización del software • optimización (por ej. ofimática) con el servidor de bases de datos
Gestión de las BBDD	<ul style="list-style-type: none"> • creación de bases de datos • importación y exportación de datos • tipos de datos y recuperación • optimización de las consultas • análisis de la integridad y reparación
Gestión de los usuarios	<ul style="list-style-type: none"> • creación de usuarios y bajas • asignación de permisos
Gestión de los recursos	<ul style="list-style-type: none"> • monitoreo del rendimiento, el tamaño • monitorización del espacio y asignación de cuotas • configuración óptima del servidor
Gestión de la seguridad	<ul style="list-style-type: none"> • monitoreo de los accesos • configuración segura del servidor

MySQL

- Relacional
- Multihilo
- Multiplataforma
- Desarrollado en C y C++
- Licencia libre y comercial

Características

- Compatible con SQL
- Arquitectura cliente/servidor
- ≠ mecanismos de almacenamiento en f/cx de las necesidades APP
- Vistas
- Proporciona procedimientos almacenados
- Disparadores
- Transacciones
- Búsqueda e indexación de rangos de texto
- APIs y librerías

Tipos de datos

- Números
 - Enteros (INTEGER, INT, INTEGER, SMALLINT, TINYINT, MEDIUMINT, BIGINT)
 - Reales (DECIMAL, NUMERIC, FLOAT, DOUBLE)
 - Bits (BIT)
- Fecha y tiempo
 - DATE, TIME, DATETIME, YEAR, TIMESTAMP
- Strings
 - Cadenas de caracteres (CHAR, VARCHAR)
 - Binarios (BINARY, VARBINARY)
 - BLOB y TEXT (TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT)
 - Enumeraciones (ENUM) – hasta 65535 cadenas de caracteres → conjunto valores posibles
 - Conjuntos (SET) – hasta 64 valores diferentes

! Largo, moroso, viene!

Sintaxis SQL

• CREATE • DROP • ALTER • RENAME TABLE • TRUNCATE TABLE

Borra todos los datos sin borrar la tabla
(más eficiente q. DELETE)

T1: ESTRUCTURA INTERNA

(MySQL)

Motores de almacenamiento

↓ tipos de tablas

cada motor utiliza su propio tipo de almacenamiento

Algunos motores:

INNODB (por defecto)

OPERACIONES DML (siguen modelo ACID):

- Atomicity → Transacciones
- Consistency → Integridad
- Isolation → IP. concurrentes
- Durability → Persistencia

- Recuperación ante errores
- InTEGRIDAD REFERENCIAL (foreign key)
- Optimizado para uso claves primarias
- Admite compresión, cifrado de datos, y réplica
- Bloqueo a nivel fila (registro) → bloquea fila mientras yo la actualizo

Indices (B-tree). Tb sobre TEXT (fulltext)

CACHE de Indices, consultas y datos

MyISAM (Indexed Sequential Access Method)

- Motor **madrugo** extensible simple de usar (no integrabilidad referencial)
- Rápido en consultas

- Soporte backups y recuperación ante errores
- Bloqueo a nivel tabla

Indices (B-tree). Tb sobre TEXT (fulltext)

CACHE de Indices, consultas y datos

- Admite compresión y cifrado de datos

cada tabla se guarda en 3 archivos:

- .frm → almacena definición de la tabla
- .MYD (MyData) → registros tabla
- .MYI (MyIndex) → índices tabla

MEMORY (HEAP)

- tablas siempre → RAM
- índices hash → acceso a registros individuales (tb B-tree)
- Acceso máx. vel. con conjunto datos pequeños

Características:

- tomar registro **FIFO**
- NO campos TIPO TEXT o BLOB
- SÍ USO AUTO_INCREMENT para columnas
- SE indices con duplicados y para columna con NULL

CSV

- datos se almacenan en .CSV [comma separate values]

Directorio storage/csv

- cada tabla se guarda como:
 - .frm → almacena formato de la tabla
 - .CSV → datos (Texto Plano)

ARCHIVE

- Pensado Grandes volúmenes no indexados
- INSERT, SELECT, UPDATE, DELETE, REPLACE
- Soporta casi todo tipo de datos (menos geoespaciales)

BLACKHOLE



- Acepta lo q llega pero no almacena nada
- todos los datos se irán a null
- Arquitecturas maestro - esclavo

Configuración del Servidor

• Fichero de config / opciones

- Windows → "my.ini" / "my.cnf"
- UNIX → "my.cnf"
- contiene las opciones de inicio para **servidor** y **clientes**
- Ficheros de texto → leidos en bloques de ignoran
- Organizado en grupos: [grupos]
- Opciones de cada grupo: opción = valor

lectura	Propósito
%PROGRAMDATA%\MySQL\Server 8.0\my.ini	Global options
%PROGRAMDATA%\MySQL\Server 8.0\my.cnf	Global options
C:\myini\my.cnf	Global options
BASEDIR\my.ini, %WINDIR%\my.cnf	The file specified with --defaults-extra-file, if any
%APPDATA%\MySQL\mylogin.cnf	Login path options (clients only)

Ficheros de log

Tipo de Log	Información escrita en el Log
Error log	Problemas encontrados durante el arranque, ejecución o parada de mysqld
General query log	Conexiones establecidas por los clientes y sentencias (consultas) recibidas de los clientes
Binary log *	Sentencias que cambian datos (también usado para replicación)
Relay log	Cambios en los datos recibidos desde un servidor maestro de replicación
Slow query log *	Consultas que tardan más de long_query_time segundos en ejecutar
DDL log (metadata log)	Operaciones sobre metadatos ejecutadas por sentencias DDL

– Por defecto, todos **desactivados** excepto "error log" en Windows

* activados en config

Algunos grupos y opciones más relevantes

client

- password = my_password
contraseña q= acceso cliente
- port = 3306
se conectan clientes (vía TCP/IP)

Mysqld

- user = mysql
usuario bajo el cual se ejecuta el servidor
- port = 3306
se establecen conexiones
- general_log = 0/no log query log
query → registrar el log de los query
- basedir = /var
directorio donde se encuentra MySQL
- datadir = /var/lib/mysql
directorio → datos (la BD)
- key_buffer_size = 16M
tamaño buffer de los índices
- max_allowed_packet = 1M
tamaño máx de un paquete (si es col. blobs)

- wait_timeout = 9600
@ espera el servidor en conexiones no interactivas

- connect_timeout = 10
@ sigue a servidor espera un paquete de conexión

- max_connections = 100
num máx de conexiones concurrentes

- max_connect_errors = 10
si entro 10 veces y te deneguen, el servidor se bloquea

- slow_query_log_file = /var/log/mysql/mysqld-slow.log
registra consultas lentas

- skip-innodb
no utiliza bd tipo INNODB

T1: ESTRUCTURA INTERNA

comandos útiles

- Arrancar el servidor (si no está como servicio o para reiniciarlo)

```
C:\> "C:\Program Files\MySQL\MySQL Server 8.0\bin\mysqld" --console
```

- Probando el servidor (si responde)

```
shell> bin/mysqladmin -u root -p version  
Enter password: (enter root password here)
```

- Variables del sistema (mysql)

```
shell> bin/mysqladmin -u root -p variables  
Enter password: (enter root password here)
```

- Parar el servidor

```
shell> bin/mysqladmin -u root -p shutdown  
Enter password: (enter root password here)
```

- Crear una base de datos

```
mysql> CREATE DATABASE nombre_bd;
```

- Borrar una base de datos

```
mysql> DROP DATABASE nombre_bd;
```

- Mostrar las tablas de la base de datos seleccionada

```
mysql> SHOW TABLES;
```

- Mostrar la estructura de una tabla

```
mysql> DESCRIBE nombre_tabla;
```

- Ver las bases de datos que existen

```
shell> bin/mysqlshow [database] -u root -p  
Enter password: (enter root password here)
```

- Conectarse al servidor

```
shell> bin/mysql [-h host] [-u user] [-p][password]
```

- Salir

```
mysql> quit | exit  
Bye
```

- Ayuda (mysql)

```
shell> bin/mysqladmin --help
```

- Introduciendo consultas (case insensitive)

```
mysql> SELECT VERSION(), CURRENT_DATE, now(), user();
```

- Consultar bases de datos existentes en el servidor

```
mysql> SHOW DATABASES;
```

- Seleccionar una base de datos

```
mysql> USE database
```

- Ver la base de datos seleccionada

```
mysql> SELECT DATABASE();
```

- Crear una tabla

```
mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),  
-> species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
```

- Borrar una tabla

```
mysql> DROP TABLE nombre_table;
```

- Cargar datos en una tabla desde un fichero de texto

```
mysql> LOAD DATA INFILE 'path/fichero.txt' INTO TABLE nombre_table;
```

- Un registro por fila
- Campos separados por TABULADORES
- Para campos con valor nulo, usar **NULL**

- Introducir un nuevo registro

```
mysql> INSERT INTO nombre_table VALUES (value1, 'value2', value3, NULL);
```

- Trabajando en **Batch Mode** (desde la línea de comandos)

```
shell> mysql [-h host -u user -p] [database] < fichero_entrada  
[> fichero_salida]
```

- Trabajando en **Batch Mode** (dentro del cliente de mysql)

```
mysql> source fichero_entrada;  
mysql> \. fichero_entrada
```